APPLICATION OF PRESSURE-PROJECTION ALGORITHMS TO A SHARP PROJECTION IMMERSED BOUNDARY METHOD FOR THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

BY

YONG YI BAY

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Aerospace Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Adviser:

Assistant Professor Vincent Le Chenadec, Director of Research

Abstract

In this thesis, we describe a globally second-order accurate sharp immersed boundary projection method with an algebraic structure parallel to the classic fractional step method for the unsteady, incompressible Navier-Stokes equations. While second-order accuracy in time and space is generally achievable for the velocity components, the pressure is usually first-order. To fully understand the source of this problem and the interplay between the pressure term and the overall fractional step method, we need to first look into the Navier-Stokes equations themselves. Perot demonstrated the possibility of higher order projection methods by means of LU approximations. This method seems applicable to any grid system and was widely accepted due to its ease of use and straightforward derivation of the order of accuracy for the pressure. It rendered another class of projection methods which relied on global pressure-updating hopeless as Strikwerda and Perot both speculated that these methods could inherently be first-order in pressure and simply cannot be improved to higher orders. Shortly after, Brown, Cortez and Minion provided insights to such pressure-updating schemes and proposed an entire class of global second-order accurate approximate projection methods. Despite Brown *et al.*'s success, it remains difficult to transfer these higher order pressure-updating methods to staggered grids. Discrete operators in staggered grids simply do not commute in the same way that colocated grids do (as in the case of approximate projection methods).

We then continue to develop suitably accurate projection approaches for the immersed boundary method (IBM). The original IB method introduced by Peskin involves solving on an Eulerian grid (usually a uniform Cartesian mesh) which does not necessarily conform to the body's geometry. Some underlying mechanism is then needed to exchange information between the flow field described by the Eulerian discretization to the set of Lagrangian points which lie on the surface of the immersed boundary. The differences between the various IB methods lie in the way they implement this mechanism. Early adaptations of the IB method required arbitrary tuning parameters either for describing the forcing effect on the immersed boundary or characterizing the boundary velocity. Taira and Colonius were the first to formulate an IB method by means of a projection approach, also known as immersed boundary projection method (IBPM), which does not

require any such parameters. Their formulation relied on Perot's LU factorization, where boundary forces and pressure terms act as Lagrange multipliers which enforce the no-slip and divergence-free constraints. This method quickly gained popularity due to its ease of use and is currently widely applied in modeling complex turbulent flows, multi-physics simulations and fluid-structure interaction. Its ability to model flow over complex geometries without the need for complex grids significantly reduced the time and effort needed for such simulations.

This thesis addresses Perot's concern of commutativity of discrete operators and proposes a way to apply Brown *et al.*'s higher-order pressure updating schemes to staggered grid arrangements. Consequently, we present a fully second-order accurate immersed boundary projection method which employs similar updating schemes that Brown uses. We also improve upon Taira and Colonius's immersed boundary projection method by using multi-linear interpolation in place of the conventional use of the discrete Dirac delta function and create a new "sharp" immersed boundary projection method.

Acknowledgments

I would like to express my sincere gratitude to Assistant Professor Vincent Le Chenadec, my adviser, for his continuous support and invaluable guidance throughout this masters program. Funding is generally difficult in the States, and I am deeply grateful to my adviser to have ever given me a chance to study at University of Illinois Urbana-Champaign. I have been very fortunate to have an adviser who sees us as co-workers with whom he collaborates with, rather than students who he advises. This has allowed me to freely voice out questions, ideas and thoughts whenever we are in a discussion. I have benefited much from his humility, patience and ever-learning attitude.

I am also indebted to Alexandru Fikl, who has allowed me to spend countless nights in his house discussing research. He has helped me overcome many crisis situations and I am truly grateful for everything that he has taught me. He has been very supportive and always guides me to recovery when my steps faltered. Words cannot express my gratitude to him.

Lastly, I would like to thank Associate Professor (Ret.) Randy Chue who has kept in contact with me over the years. His personal stories and encouragement has been critical in my development as a graduate student as I overcome the steep learning curve. I am grateful for his mentorship and deeply appreciative of his faith in me. I have been very blessed to have him as my adviser in my undergraduate days and a close friend now.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

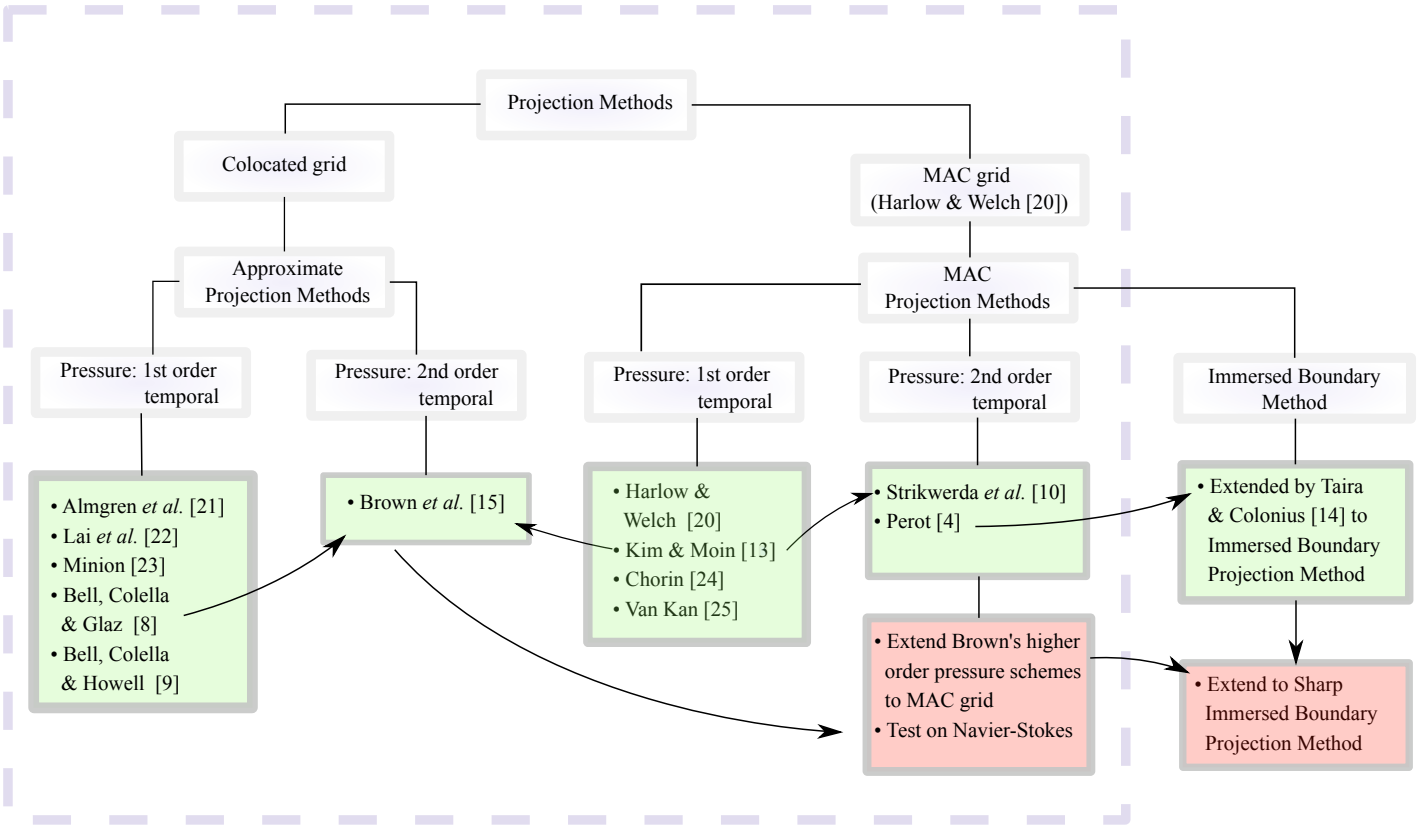


Figure 1.1: Main developments in projection methods. Legend:
"Arrows": All arrows in the flow chart should be understood as "improved upon". For instance, the box "Brown *et al.*" has arrows pointing at it from "Bell, Colella & Glaz" and "Kim & Moin". It should be interpreted as "Brown *et al.*" improved upon both "Bell, Colella & Glaz", and "Kim & Moin's" first-order pressure schemes to achieve second-order temporal schemes.
"Green" color boxes: These are papers that have already been published and should be thought of as present knowledge.
"Red" color boxes: These are our modifications and contributions.
"Dotted" box: Everything within the dotted box is within the context of Navier-Stokes equations.

1.1 Background

The topic of projection methods and the resulting fractional step methods has been a topic of considerable debate and interest since its inception in 1968 [24]. As of now, there have been various classes of projection methods where numerous authors seek higher order accuracy in vastly different ways. While second-order temporal accuracy is usually achievable for the velocity, the pressure term in the Navier-Stokes equations is usually first-order in time even when a second-order integration technique is used. In this thesis, we seek to explain the oddities of the pressure term and the role it plays as a Lagrange multiplier.

The ability or possibility of achieving higher order accuracy for the pressure is imperative in the work towards the development of accurate immersed boundary projection methods. In the scope of immersed boundary projection methods, there exists more than one Lagrange multiplier (such as boundary forces, etc.) and, if we understand how to achieve higher order accuracy in the pressure term in Navier-Stokes, we can gain valuable insight into developing more general second or higher order accurate immersed boundary projection methods.

1.2 Origin of Projection Methods

Given an n-dimensional bounded domain Ω with boundary $\partial\Omega$, the incompressible Navier-Stokes equations with appropriate initial and boundary conditions are:

$$\begin{aligned} &\mathbf{u}_t + \nabla p = -(\mathbf{u}\cdot\nabla)\mathbf{u} + \nu\nabla^2\mathbf{u}, \\ &\nabla\cdot\mathbf{u} = 0, \end{aligned} \qquad (\mathbf{x},t) \in \Omega\times[0,T], \tag{1.1}$$

$$\mathcal{B}(\mathbf{u},p) = 0, \qquad (\mathbf{x},t) \in \partial\Omega\times[0,T], \tag{1.2}$$

$$\mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}), \qquad \mathbf{x}\in\Omega. \tag{1.3}$$

where $\mathbf{u} \equiv \mathbf{u}(\mathbf{x},t)$ is the fluid velocity field, $p \equiv p(\mathbf{x},t)$ is the pressure, ν is the kinematic viscosity and $\mathcal{B}(\mathbf{u},p) = 0$ consists of n boundary conditions.

The original projection method developed by Chorin [24] in 1968 is based on the discrete Hodge decomposition. Chorin recognized that the left-hand side of equation (1.1) together with $(\mathbf{u}\cdot\hat{\mathbf{n}}) = 0$ on the domain boundary $\partial\Omega$ represents a Hodge decomposition. The interested reader should refer to Minion [23] for the derivation and discussion of the continuous and discrete Hodge decomposition. Hodge decomposition (in the simplest form) states that any vector field can be uniquely decomposed into a solenoidal (or divergence-free) vector

field and the gradient of a scalar. As such the resulting projection method first calculates an intermediate velocity field then project it onto the divergence-free fields to recover the solenoidal velocity.

Chorin's projection method requires the discrete divergence operator D and discrete gradient operator G to be skew-adjoint; i.e.

$$D = -G^T \tag{1.4}$$

and the projection $\mathbf{P}$ to be defined as

$$\mathbf{P} = I - G(DG)^{-1}D. \tag{1.5}$$

The adjointness of these discrete operators determines the uniqueness of the resulting solution. Chorin presented the error estimates of his projection method [28] in a follow up to his original paper [24] and showed that his method is first-order accurate in time and second-order in space for a periodic box in both 2-D and 3-D. Since Chorin was implementing the projection method on a staggered-grid system, complications arise as the projection does not commute with the Laplacian in the viscous term in presence of boundaries. In [24, 28], Chorin proposed a fix by using inhomogeneous boundary conditions when solving for the intermediate velocity field. This was an important milestone as Kim and Moin [13] subsequently adopted this method in their implementation of the projection method.

1.3 Previous Works

Approximate Projection Methods

Approximate projection methods satisfy the discrete divergence-free condition up to the truncation error:

$$D^{MAC}\mathbf{u} = \mathcal{O}(h^2) \tag{1.6}$$

where h is the mesh-spacing and D^{MAC} is the divergence operator pertaining to the MAC (marker-and-cell) projection which will be detailed later.

Bell, Colella and Glaz [8] came up with a promising scheme which combines the concept of Godunov's method (for discretizing the non-linear convective term) together with projection methods. Unfortunately, the matrimony between the two was a strained one. Godunov's method required velocity components u and v to be at cell-centers due to the cell-centered discretization inherent in the method, whereas a MAC projection method satisfies the continuity constraint to machine precision due to location of velocity at cell-edges (in

2-D) or cell-faces (in 3-D). We will demonstrate this issue by presenting a simple 2-D toy example with simple centered-differencing schemes. The divergence operator using centered differencing is:

$$(D^c\mathbf{u})_{ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} + \frac{v_{i,j+1} - v_{i,j-1}}{2h} \tag{1.7}$$

and the gradient operator is similarly given by:

$$(G^c\phi)_{ij} = \left(\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h}, \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2h}\right), \tag{1.8}$$

where ϕ is the solution to the discrete Poisson equation. We recall that in the Poisson equation present in the fractional step, we have to invert a Laplacian operator D^cG^c:

$$(D^cG^c\phi)_{ij} = \frac{\phi_{j+2,j} + \phi_{i+2,j} - 4\phi_{ij} + \phi_{i-2,j} + \phi_{i,j-2}}{4h^2}. \tag{1.9}$$

This looks similar to the "star-stencil" that we usually see in a five-point Laplacian, except for the fact that it is twice as wide. The stencil in Eq. (1.9) has a spacing of $2h$ and it decouples onto four distinct subgrids where values on each subgrid are independent of the others. This means that every point relies only on other values on their own subgrids (except possibly at boundaries), and hence is a locally decoupled stencil, as coined by Almgren [21]. Minion [23] explains that in the periodic boundary case, this results in an oscillatory mode in the null space of the Laplacian operator, where the null space is a sum of the different constants in each subgrid (checkerboard effect). This oscillatory mode is unfortunately also present in the divergence operator. Minion pointed that in the presence of sharp gradients in pressure or velocity, this mode can be introduced into the velocity field and there might be no way for the projection to remove it. This is also noted by Almgren *et al.* [21] where the authors commented that the local decoupling results in a 2^d-dimensional kernel for G^c where d is the dimension of the problem.

Bell, Colella and Howell [9] came up with a fix for this problem shortly after. They recommended the use of multigrid method to solve the Poisson equation. Although such multigrid methods fit nicely into the concept of locally refined grids, it was proven to be difficult to implement by Howell [26]. The need for the multigrid operator to maintain the decoupling of Laplacian stencil is pretty involved and arduous. Lai, Bell and Colella [22] also came up with a "filtering" technique to filter these oscillations present in the velocity field at every time step. They presented the method to be successful in the simulation of reacting flows. Rhie and Chow [1] also came up with a solution for this checkerboard effect by adding a pressure gradient (evaluated at the face) to the intermediate velocity field (interpolated to the face). This method widely known as the Rhie-Chow Interpolation is essentially adding a term that is proportional to the third derivative of the pressure. This term is proportional to a fourth order derivative in the continuity equation and acts as

dissipative or damping term which suppresses the pressure oscillations.

One other way of avoiding the decoupled Laplacian stencil is to use a MAC projector [9]. A MAC projector is the projection that one would use when using a MAC grid. In other words, we are using a MAC projection on a colocated grid (this is also known as approximate projection). Since the velocity components lie at cell centers, interpolation of the velocity to the cell faces will be necessary. For example, using:

$$\tilde{u}_{i-1/2,j} = \frac{u_{i-1,j} + u_{i,j}}{2} \quad \text{and} \quad \tilde{v}_{i,j-1/2} = \frac{v_{i,j-1} + v_{i,j}}{2}. \tag{1.10}$$

The resulting discrete Poisson equation to be solved is:

$$(D^{MAC} G^{MAC} \phi)_{ij} = (D^{MAC} \tilde{\mathbf{u}})_{ij} \tag{1.11}$$

and solutions will have to be interpolated from cell faces back to cell centers. Minion [23] states that the main advantages of approximation projection methods includes: (1) easy extension to locally refined grid implementation, (2) guaranteed well-posedness since MAC projection is already well-posed and (3) unambiguous application of boundary conditions for the approximate projection Poisson problem in Eq. (1.11).

MAC Projection Methods

MAC projection methods satisfy the discrete divergence-free condition up to machine precision ϵ:

$$D^{MAC} \mathbf{u} = \mathcal{O}(\epsilon). \tag{1.12}$$

The staggered grid system or MAC grid first sees an application in Harlow and Welch's numerical simulations for incompressible flow [20]. On the MAC grid, vector quantities such as velocity components u and v are stored on cell edges in 2-D (cell faces in 3-D) while scalar quantities such as the divergence of a vector field or the pressure are represented at cell centers.

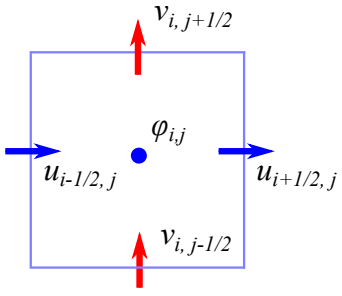


Figure 1.2: Variable arrangement on a staggered grid.

We define the MAC divergence as follows:

$$(D^{MAC}\mathbf{u})_{i,j} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{h} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{h}. \tag{1.13}$$

It is important to note that the MAC divergence is conservative and, thus, the discrete analog of the divergence theorem is trivially true. Minion [23] derived two very important properties of the MAC projection in his derivation of the discrete Hodge decomposition: (1) the existence of a unique solution depends on the adjointness or orthogonality condition of the discrete operators D^{MAC} and G^{MAC}, (2) the projection $\mathbf{P}$ whose definition is given in [23, 21] has the norm reducing property and hence is a stable method. Lastly, one can see that the resulting Laplacian on a MAC grid uses the familiar 5-point stencil:

$$\left(D^{MAC}G^{MAC}\phi\right)_{i,j} = \frac{\phi_{j+1,j} + \phi_{i+1,j} - 4\phi_{ij} + \phi_{i-1,j} + \phi_{i,j-1}}{h^2}. \tag{1.14}$$

Kim and Moin [13] implemented the MAC projection method using explicit Adams–Bashforth for the non-linear convective terms and Crank-Nicolson for the viscous term. Numerical evidence showed their scheme to be second-order accurate. They took up Chorin's [24, 28] technique of tackling problems with boundaries and applied inhomogeneous boundary conditions for solving the intermediate velocity field:

$$\mathbf{u}^* = \mathbf{u}^{n+1} + \Delta t \nabla \phi^n, \text{ on } \partial\Omega. \tag{1.15}$$

This intermediate boundary condition uses a lagged value of ϕ^n to approximate ϕ^{n+1} which is not available at this intermediate step. Brown *et al.* [15] later proved in their paper, using normal mode analysis, that this is in fact a sufficient condition to obtain second-order accuracy in the velocity components and is also a necessary condition to achieve global second-order for the scheme. Kim and Moin also pointed out that their auxiliary pressure variable ϕ (which is also the solution to the Poisson problem) can be used to recover the pressure variable in the following way:

$$p = \phi - \nu \frac{\Delta t}{2} \nabla^2 \phi.$$

It is not well understood what time-step the pressure p or ϕ is in fact evaluated at and this has resulted in considerable debate in the literature by authors such as Perot [4], Strikwerda and Lee [10]. If the pressure variable and ϕ were both evaluated at the same time $n+1$, then Strikwerda and Lee proved that the pressure is only first-order accurate in time. This is one of the main reasons why Perot and Strikwerda speculated that fractional step methods as such are inherently first-order in pressure and cannot be improved to achieve higher order in time.

However, Brown *et al.* showed that, in order for Kim and Moin's fractional step to be consistent with the

second-order discretization of the Crank-Nicolson method, the pressure recovery equation in Kim and Moin's paper has to be interpreted as the time-centered pressure:

$$p^{n+1/2} = \phi^{n+1} - \frac{\nu \Delta t}{2} \nabla^2 \phi^{n+1}. \tag{1.16}$$

Brown *et al.* demonstrates, using normal mode analysis, that Eq. (1.16) does indeed lead to global second-order accuracy in time. Van Kan [25] subsequently proposed a second-order generalization of such projection methods. His scheme is first discretized in space using the staggered grid of Harlow and Welch and in time using a second-order time integration. Perot provided numerical evidence that the trapezoidal pressure scheme proposed by van Kan is in fact first-order in time even though the time integration technique was supposedly second-order. Bell *et al.* [8] also commented that van Kan's analysis assumes fixed bounds on the operators and is hence invalid if space and time were refined simultaneously.

Immersed Boundary Projection Methods

The immersed boundary method (IBM) was originally described by Peskin [34] in 1972 with an application to the simulation of blood flow in heart valves. The implementation involved the spatial discretization using an Eulerian grid with a set of Lagrangian points on the surface of the immersed boundary. The main advantage of the method came from the fact that the Eulerian discretization did not have to conform to the geometry of the immersed body. No-slip conditions were enforced on the immersed surface by adding appropriate boundary forces at the Lagrangian points. In the continuous equations, these boundary forces are singular functions which exist on the immersed surface and are described by discrete Dirac Delta functions which "smear" the forcing effect to neighboring Eulerian cells. Peskin proposed the forcing function by means of Hooke's Law. In 1992, Beyer and LeVeque [36] subsequently used a spring constant $\gg 1$ to simulate flow over rigid bodies. They noted that naive approximations of the Dirac Delta function can affect the order of convergence and correction terms are sometimes required to maintain accuracy levels. Goldstein *et al.* [35] then proposed the use of feedback control instead of the conventional use of Hooke's Law for the forcing function in 1993. In their method, velocity is iterated using a proportional-integral control to determine the desired velocity at the immersed boundary.

The gain in the feedback control or the spring constant (or stiffness) in Hooke's Law all seemed arbitrary in calculating the forces to be applied on the immersed boundary. Taira and Colonius [14] in 2007 sought to disregard all these tuning parameters and proposed a projection approach for the IB Method. The immersed

boundary projection method (IBPM) regards boundary forces and pressure as Lagrange multipliers that enforce the no-slip boundary condition on the immersed surface and divergence-free constraint, respectively. The authors closely followed Perot's [4] algebraic approach towards the Navier-Stokes equations and applied a block LU factorization to the resulting IBPM equations in order to avoid the solution of a fully coupled system (saddle point problem). IBPM garnered popularity quickly as its IB capability can be easily added to an existing Navier-Stokes solver which employs a projection method [37].

1.4 Scope and Outline of the Present Work

Navier-Stokes Equations

It might seem trivial at first to extend Brown *et al.*'s [15] global second-order pressure update schemes to the staggered-grid system. However, there problem is considerately more complex than can be deduced at first sight. Brown *et al.*'s higher order update schemes are able to achieve global second-order accuracy because of a very important ingredient which is absent in other works, which use staggered grids. The colocation of velocity components allowed certain discrete operators (viscous and gradient) to commute and hence allowed for the possibility of higher order pressure update schemes. However, is it possible to stretch this to the staggered-grid system, where discrete operators are not square and do not commute? We will show that, even when the operators do not commute, a second-order approximation of the updated pressure can be obtained if an additional Poisson equation is solved. It should be noted however that this extra step involves a significant computational overhead. The details of which are slightly involved and will be explained in a coherent manner in subsequent chapters.

Immersed Boundary Projection Methods

A thorough understanding of the projection method for the Navier-Stokes equations is critical in extending existing results to the immersed boundary projection method (IBPM). In the Navier-Stokes equations, pressure is seen as a Lagrange multiplier to enforce the divergence-free constraint and, similarly, in IBPM, boundary forces applied at immersed surfaces are used as Lagrange multipliers to enforce the no-slip constraint. The extension to accurate IBPMs is straightforward as long as we learn the ways of achieving higher order accuracy for the pressure term in Navier-Stokes. As we see it, there are two distinct methodologies applied

to this problem: (1) Perot's [4] LU approximation and (2) Brown *et al.*'s [15] higher order pressure update schemes. These two methodologies allow for the possibility of higher order temporal accuracy in pressure and are suitable candidates to be used in IBPM. Taira and Colonius [14] have demonstrated the use of Perot's LU approximation to achieve global second-order IBPM schemes. In this thesis, we attempt to explore the second route, using Brown's improved fractional step methods for use in IBPM.

Next, we conduct a thorough derivation of Taira and Colonius's IBPM to understand their implementation in sufficient detail. A derivation of the Karush–Kuhn–Tucker conditions revealed intricate technicalities which explained their choice of solving in velocity fluxes instead of primitive variables and so on. In this thesis, we formulate a sharp immersed boundary projection method where we solve in primitive variables (i.e. velocity) and replace the discrete Dirac Delta function by a sharper multi-linear interpolation. Taira and Colonius used the discrete Dirac Delta function [27] in their implementation, which involved the smearing of boundary forces from the Lagrangian points (which lie on the immersed boundary surface) to neighboring cells. If multi-linear interpolation were to be used, there will be no smearing of forces and hence the "sharp" characteristic. Taira and Colonius also created colocated Lagrangian points where the (x, y, z) components of the forces all lie at the same point. We will however have one set of Lagrangian points per dimension, with each Lagrangian point chosen to be the center of the immersed segment within each of the staggered lattices.

Part I

Navier-Stokes Equations

Chapter 2

Survey of Projection Methods

2.1 Introduction

A survey on various fractional step methods will be critical to our understanding of the immersed boundary projection method. It has been proven in many highly cited papers that the fractional step method results in first-order temporal accuracy in pressure with second-order accuracy for the velocity [8, 10, 11, 15]. This is important because pressure acts as a Lagrange multiplier and we will eventually increase the number of Lagrange multipliers in the immersed boundary method. Hence, analyzing the possibility of achieving second-order accuracy in pressure is a first step in understanding how to achieve higher order for all such multipliers.

Let's review the fractional step method for the Stokes equations in a more general manner as presented by Brown [15]. The method has 3 steps:

Step 1: Solve for the intermediate velocity $\mathbf{u}^*$:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \nabla q = \frac{\nu}{2}\nabla^2(\mathbf{u}^* + \mathbf{u}^n),$$

where ∇q is an approximation to the pressure gradient $\nabla p^{n+1/2}$ used in finding the intermediate velocity.

Step 2: Perform a projection to solve for $\mathbf{u}^{n+1}$:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla \phi^{n+1}.$$

Step 3: Update the pressure:

$$p^{n+1/2} = q + \mathbb{L}\phi^{n+1},$$

where $\mathbb{L}$ is a linear differential operator.

In the following sections, we survey a few of the most important references in the context of second-order

projection methods. For simplicity, we will restrict the discussion to the Stokes equations. The highlight would be Perot's [4] LU approximation, where no pressure update is required, and the various pressure update schemes proposed by Bell, Colella and Glaz [8] and Kim and Moin [13].

2.2 Approximate LU Factorization Methods

The fully discretized Stokes equations are:

$$\begin{cases} \dfrac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t} + (A\ B^N)Gp^{n+1} = \dfrac{\nu}{2}L(\mathbf{u}^{n+1}+\mathbf{u}^n) + \boldsymbol{bc}_1, \\ D\mathbf{u}^{n+1} = bc_2, \end{cases} \tag{2.1}$$

where L is the discrete Laplacian, G is the discrete gradient, D is the discrete divergence and

$$A = \left(\frac{1}{\Delta t} - \frac{\nu}{2}L\right), \tag{2.2a}$$

$$B^N \approx A^{-1} = \Delta t + \frac{\Delta t^2}{2}L + \cdots + \frac{\Delta t^N}{2^{N-1}}L^{N-1}, \tag{2.2b}$$

with B^N as an N^{th} order approximation of the inverse of A. Using this specific discretization, the steps of the method can be reformulated as follows:

Step 1: Solve for the intermediate velocity $\mathbf{u}^*$:

$$\frac{\mathbf{u}^*-\mathbf{u}^n}{\Delta t} = \frac{\nu}{2}L(\mathbf{u}^*+\mathbf{u}^n) + \boldsymbol{bc}_1. \tag{2.3}$$

Step 2: Perform the projection to find the divergence-free velocity $\mathbf{u}^{n+1}$:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - B^N Gp^{n+1}, \tag{2.4}$$

where

$$DB^N Gp^{n+1} = D\mathbf{u}^* - bc_2. \tag{2.5a}$$

By substituting $\mathbf{u}^*$ from Eq. (2.4) into Eq. (2.3) we can get back our original governing equations in Eq. (2.1), so this method is a valid approximation. It can be shown that Eq. (2.1) is obtained from an LU approximation.

Analog to LU Approximation

Re-writing Eq. (2.1) in block matrix form, we have:

$$\begin{pmatrix} A & (AB^N)G \\ D & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{r}^n \\ 0 \end{pmatrix} + \begin{pmatrix} \boldsymbol{b}c_1 \\ bc_2 \end{pmatrix} \tag{2.6}$$

The operator acting on the concatenated vector $(\mathbf{u}^{n+1}, p^{n+1})$ can be decomposed using LU factorization as follows:

$$\begin{pmatrix} A & 0 \\ D & -DB^NG \end{pmatrix} \begin{pmatrix} I & B^NG \\ 0 & I \end{pmatrix} \begin{pmatrix} \mathbf{u}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{r}^n \\ 0 \end{pmatrix}, \tag{2.7}$$

where

$$\boldsymbol{r}^n = \left(\frac{1}{\Delta t} + \frac{\nu}{2} L \right) \mathbf{u}^n.$$

Reconciling Perot's LU approximation with Brown's higher order pressure-updating schemes has been a challenging endeavor. As can be seen, in the realm of LU approximations, there is no such thing as a pressure update. This is because the pressure update step simply cannot be factored into the discrete operators in Eq. (2.7). Perot's LU approximation might just be a very different method of achieving higher order temporal convergence for the pressure, where the order of convergence of the pressure is determined by the approximation of A^{-1} in Eq. (2.2b). At this point, we should reiterate a point made by Perot in [5], where he stated that his method of achieving higher order convergence for the pressure is by no means the only way. There are possibilities of improving accuracy by using higher order boundary conditions such as in the method presented by Brown [15]. Perot stated that his method is merely straightforward as it is easy to derive the order of accuracy from A^{-1}, whereas Brown actually had to use normal mode analysis to arrive at his order of accuracy.

2.3 Incremental-Pressure Projection Methods

In the case of incremental-pressure projection methods, we start with the semi-discrete Stokes equations (discrete in time, continuous in space), which are:

$$\begin{cases} \dfrac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \nabla p^{n+1/2} = \dfrac{\nu}{2} \nabla^2 (\mathbf{u}^{n+1} + \mathbf{u}^n), \\ \nabla \cdot \mathbf{u}^{n+1} = 0. \end{cases} \tag{2.8}$$

Bell *et al.* [8] derived that their pressure update scheme results in second-order accurate velocity while retaining first-order accuracy for the pressure. This is very much in line with what Perot [4] mentioned. An improvement on their pressure update scheme will be proposed later in this section and this modification is later confirmed to bring about second-order temporal accuracy in pressure using normal mode analysis in Section 3.3. Let us take a look at Bell *et al.*'s fractional step method:

Step 1: Solve for the intermediate velocity $\mathbf{u}^*$:

$$\begin{aligned} &\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \nabla p^{n-1/2} = \frac{\nu}{2}\nabla^2(\mathbf{u}^* + \mathbf{u}^n), \\ &\mathcal{B}(\mathbf{u}^*) = \left(\mathbf{u}^* - \mathbf{u}^{n+1}\right)|_{\partial\Omega} = 0. \end{aligned} \tag{2.9}$$

Step 2: Perform the projection to obtain the divergence-free velocity $\mathbf{u}^{n+1}$:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla \phi^{n+1}, \tag{2.10}$$

where

$$\Delta t \nabla^2 \phi^{n+1} = \nabla \cdot \mathbf{u}^*, \tag{2.11a}$$

$$\hat{\mathbf{n}} \cdot \nabla \phi^{n+1}|_{\partial\Omega} = 0. \tag{2.11b}$$

Step 3: Update the pressure:

$$p^{n+1/2} = p^{n-1/2} + \phi^{n+1}. \tag{2.12}$$

Kim *et al.* [7] employed a "delta form" method and employed the following pressure-updating scheme $p^{n+1/2} = p^{n-1/2} + \delta p$. We noticed the equivalence of this pressure-update scheme with Bell *et al.*'s where δp is really just ϕ^{n+1}. Brown *et al.* advised that this pressure update scheme is not consistent with a second-order discretization of the Navier–Stokes equations. One can easily check this by substituting $\mathbf{u}^*$ from Eq. (2.10) into Eq. (2.9) and comparing with Eq. (2.8). The result is as follows:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \underbrace{\nabla p^{n-1/2} + \nabla \phi^{n+1} \overbrace{- \frac{\nu \Delta t}{2} \nabla^2 \left(\nabla \phi^{n+1}\right)}^{\text{missing term}}}_{\approx \nabla p^{n+1/2}} = \frac{\nu}{2}\nabla^2(\mathbf{u}^{n+1} + \mathbf{u}^n). \tag{2.13}$$

There's a missing term in the pressure update scheme in Eq. (2.12) and hence the inconsistency. From Eq. (2.13), the correct pressure gradient update that is consistent with a second-order Crank-Nicolson method should be:

$$\nabla p^{n+1/2} = \nabla p^{n-1/2} + \nabla \phi^{n+1} - \frac{\nu \Delta t}{2} \nabla^2 \left(\nabla \phi^{n+1}\right). \tag{2.14}$$

2.3.1 Complications in the Discrete Analog

The above discussion applies to the context of the implementation of fractional step methods for a semi-discrete Stokes equation (continuous in space and discrete in time). Would Eq. (2.14) still work when Stokes equations are fully discretized? Can we say that:

$$\nabla_h p^{n+1/2} = \nabla_h p^{n-1/2} + \nabla_h \phi^{n+1} - \frac{\nu \Delta t}{2} \nabla^2_{vis} \left(\nabla_h \phi^{n+1} \right)$$

and hence:

$$p^{n+1/2} = p^{n-1/2} + \phi^{n+1} - \frac{\nu \Delta t}{2} \nabla^2_{vis} \phi^{n+1} ? \tag{2.15}$$

Eq. (2.15) is wrong *in general* (the choice of this wording will be clear shortly). Brown's normal mode analysis is done on the continuous operators and thus commutativity is not an issue. However, Brown seemed to have evaded the problem of commutativity in discrete operators and extended this naturally to the numerical tests and presented the following pressure gradient update:

$$\nabla_h p^{n+1/2} = \nabla_h p^{n-1/2} + \nabla_h \phi^{n+1} - \frac{\nu \Delta t}{2} \nabla_h (\nabla^2_{vis} \phi^{n+1}). \tag{2.16}$$

We would like to know if there are exceptions to this problem of commutativity. There are two known cases where the two operators commute, namely:

Periodic Boundary Conditions. Perot [5] mentioned in his paper that "...the only exception to this rule known to the author is periodic boundaries." This would mean that the gradient and viscous operators commute in the periodic case. This is something that we would like to look into (at least in the numerical case) in future implementations.

Colocated Grids. This is the main reason why we believe Eq. (2.16) might actually be true in Brown *et al.*'s case. Only in the colocated grid arrangement, can the discrete operators commute in the way suggested by Brown. Although Brown [15] did not mention the grid arrangement used in his numerical tests, there are some signs that suggest so.

In the colocated system, the velocity components u_i (or u, v, w) and the pressure are all stored at the cell centers. Volume fluxes are defined at cell faces similar to that of the staggered-grid arrangement. These volume fluxes are calculated by means of an interpolation of u_i followed by a projection operation which guarantees exact mass conservation as noted by Rhie and Chow [1] in 1983. Felten and Lund [3] did a study on the comparison of colocated and staggered grid arrangements for incompressible turbulent

flows and noted that the major draw-back from such a method is that the cell-centered velocity are only approximately divergence free. This might explain why there's a table in Brown *et al.*'s paper where they measure the numerical errors for the divergence of velocity field. One of the biggest give-away that truly showed the use of colocated grids in Brown's numerical tests, is the lack of distinction in ∇_h^2 in the discrete Poisson problem in Eq. (2.11a) and the viscous operator in Eq. (2.9). In Brown's paper, both operators are written as ∇_h^2 in the discrete analog. If they were in staggered-grid system, these 2 discrete operators would have required distinction since the viscous term acts on face-centered values while the Laplacian operator in discrete Poisson problem acts on cell-centered values.

The staggered grid arrangement, when formulated properly, is able to conserve mass, momentum and energy regardless of underlying coordinate system, as investigated by Morinishi *et al.* [2]. Its conservation properties are attractive and they are the main reason why we'll be employing the staggered grid arrangement, as do many other researchers [4, 8, 9, 13].

2.3.2 Solution to the Problem of Commutativity

$$\nabla_h p^{n+1/2} = \underbrace{\nabla_h p^{n-1/2} + \nabla_h \phi^{n+1} - \overbrace{\frac{\nu \Delta t}{2} \nabla_{vis}^2 \nabla_h \phi^{n+1}}^{(*)}}_{RHS} \tag{2.17}$$

Eq. (2.17) is certainly consistent with the second-order discretization of Navier-Stokes equations. However, due to lack of commutativity in general, we can't just remove the gradient operator. If ∇_h and ∇_{vis}^2 do not commute, it brings about another question as to whether the term in $(*)$ belongs to the column space of ∇_h. The aim is to extend this implementation to staggered grid systems and our solution to this problem is a simple one. We noted that within the fractional step, pressure is never needed explicitly in time stepping. We only require the gradient of pressure to march in time. As such, Eq. (2.17) will be used for the pressure gradient updates. At the very end of the simulation, if we would like to obtain the pressure variable, we can take a divergence of Eq. (2.17) and invert the cell-centered Laplacian.

$$\nabla_h^2 p^{n+1/2} = \nabla_h \cdot (RHS) \tag{2.18}$$

2.3.3 Summary of the Improved Fractional Step

The steps of the proposed fractional step methods are:

Step 1: Solve for intermediate the velocity $\mathbf{u}^*$:

$$\begin{aligned} &\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \nabla_h p^{n-1/2} = \frac{\nu}{2}\nabla^2_{vis}(\mathbf{u}^* + \mathbf{u}^n), \\ &\mathcal{B}(\mathbf{u}^*) = \left(\mathbf{u}^* - \mathbf{u}^{n+1}\right)|_{\partial\Omega} = 0. \end{aligned} \tag{2.19}$$

Step 2: Perform projection to obtain divergence-free velocity $\mathbf{u}^{n+1}$:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla_h \phi^{n+1}, \tag{2.20}$$

where

$$\Delta t \nabla^2_h \phi^{n+1} = \nabla_h \cdot \mathbf{u}^*, \tag{2.21a}$$

$$\hat{\mathbf{n}} \cdot \nabla \phi^{n+1}|_{\partial\Omega} = 0. \tag{2.21b}$$

Step 3: Update the pressure gradient:

$$\nabla_h p^{n+1/2} = \nabla_h p^{n-1/2} + \nabla_h \phi^{n+1} - \frac{\nu \Delta t}{2}\nabla^2_{vis}\nabla_h \phi^{n+1}. \tag{2.22}$$

Step 4 (Optional): Retrieving the pressure by inverting the cell-centered Laplacian:

$$\nabla^2_h p^{n+1/2} = \nabla_h \cdot \left(\nabla_h p^{n-1/2} + \nabla_h \phi^{n+1} - \frac{\nu \Delta t}{2}\nabla^2_{vis}\nabla_h \phi^{n+1}\right). \tag{2.23}$$

2.4 Pressure-Free Projection Methods

The projection methods mentioned in Kim *et al.* [7], and Bell *et al.* [8] are called *pressure increment projection methods*, where $q \neq 0$. Kim and Moin's [13] fractional step method is part of the *pressure-free methods*, where $q = 0$. We pick up the semi-discrete Stokes equation as given by Eq. (2.8). In this case, the fractional step method looks like:

Step 1: Solve for the intermediate velocity $\mathbf{u}^*$:

$$\begin{aligned} &\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \frac{\nu}{2}\nabla^2(\mathbf{u}^* + \mathbf{u}^n), \\ &\mathcal{B}(\mathbf{u}^*) = \mathbf{u}^*|_{\partial\Omega} = \left(\mathbf{u}^{n+1} + \Delta t \nabla \phi^n\right)|_{\partial\Omega}. \end{aligned} \tag{2.24}$$

Step 2: Perform the projection to obtain the divergence-free velocity $\mathbf{u}^{n+1}$:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla \phi^{n+1}, \tag{2.25}$$

where

$$\Delta t \nabla^2 \phi^{n+1} = \nabla \cdot \mathbf{u}^*, \tag{2.26a}$$

$$\hat{\mathbf{n}} \cdot \nabla \phi^{n+1}|_{\partial\Omega} = 0. \tag{2.26b}$$

Step 3 (Optional): Retrieving the pressure variable:

$$p^{n+1/2} = \phi^{n+1} - \frac{\nu \Delta t}{2} \nabla^2 \phi^{n+1}. \tag{2.27}$$

Substituting $\mathbf{u}^*$ from Eq. (2.25) into Eq. (2.24) and comparing with Eq. (2.8) reveals that this pressure update is consistent with the second-order Crank-Nicolson method.

As is often the case, in this method the pressure is not explicitly required to advance velocity. The auxiliary pressure variable ϕ is also discarded at the end of every time step. Kim and Moin's fractional step seemed to be consistent with the second-order Crank-Nicolson method. However, does it necessarily lead to second-order temporal accuracy? As it turns out, it really depends. Kim and Moin presented the pressure equation to be $p = \phi - (\nu \Delta t)/2)\nabla^2 \phi$ without denoting the time at which both p and ϕ are evaluated at. If the pressure variable and ϕ were both evaluated at same time $n + 1$, as assumed by Strikwerda and Lee [10], then the answer is no. Their analysis shows that boundary conditions simply cannot be satisfied in the projection step. The original governing equation Eq. (2.8) requires two boundary conditions while the system of equations Eqs. (2.24) and (2.26a) inherent in the method only requires one boundary condition. Strikwerda *et al.* showed that the inability of the method to satisfy the boundary conditions limited the accuracy of the fractional step. The result is second-order for the velocity and first-order for the pressure.

Brown *et al.* provided an alternative look at Kim and Moin's fractional step method. Brown noted that the pressure recovery equation should be understood to be $p^{n+1/2} = \phi^{n+1} - (\nu \Delta t)/2)\nabla^2 \phi^{n+1}$ for consistency of the second-order discretization of Navier-Stokes equations. Brown proved his claim by normal mode analysis and also provided numerical evidence that higher-order accuracy boundary conditions (more specifically at least first-order) is needed to achieve second-order accuracy in pressure. The possibility of using higher order boundary conditions to improve the order of accuracy of pressure in fractional step methods is also mentioned by Perot [5].

2.5 Summary

This section reviewed some of the most influential papers in fractional step analysis. We have seen that Perot's [4] LU approximation allows for higher order accuracy in the pressure term by means of higher order approximation of A^{-1}. In fact, the temporal accuracy of the pressure relates directly to the order of accuracy of the approximation to A^{-1} that is used. As such, the method makes it easy to derive the order of accuracy of the resulting scheme. The pressure increment projection methods, such as that of Bell, Colella and Glaz [8], however require the use of normal mode analysis [10, 15] or even more contrived methods employed by E and Liu [11] to arrive at the order of convergence of the resulting fractional step method. We explained the expectations of a fractional step method to be consistent to a second-order discretized Stokes equation, and noted the flaw in Bell, Colella and Glaz's pressure update scheme. Brown [15] suggested changes to that scheme to achieve global second-order accuracy which appeared to be only applicable to the colocated grid system. The major highlight in this section is our proposal of an improved incremental pressure projection method for staggered grids.

Chapter 3

Normal Mode Analysis

By using transforms for analyzing difference schemes and differential equations [12], we are going to prove the numerical oddities of the pressure and its role in the fractional step methods. Normal mode analysis will first reveal the spurious mode that exists in certain pressure update schemes, and we'll then show that this spurious modes will subsequently affect the order of convergence of the pressure but never the velocity. The appropriate modifications that have to be made to such schemes are also analyzed. The following analysis is conducted on the Stokes equations.

We conducted a survey in Chapter 2 on fractional step methods and revealed their intuitive consistency to a second-order discretized system. We now go on a more rigorous path in section 3.2.1 to analyze the various pressure update schemes using normal mode analysis and show that some pressure update schemes result in spurious modes. In Section 3.1 we derive the "reference" solution for the Stokes equation. We then derive the solutions for the semi-discrete Stokes equation (continuous in space, discrete in time) and compare it with the reference solution to attain the expected order of convergence of the velocity and the pressure variables. Consequently, the suitable update scheme, which will be of second-order convergence, will be used for our Lagrange multipliers in the immersed boundary projection method.

Let us restate Brown *et al.*'s generalization of projection methods for the Stokes equations:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \nabla q = \frac{\nu}{2}\nabla^2(\mathbf{u}^* + \mathbf{u}^n), \tag{3.1}$$

where ∇q is an approximation to the pressure gradient $\nabla p^{n+1/2}$ used in finding the intermediate velocity. The velocity satisfies $\nabla \cdot \mathbf{u}^{n+1} = 0$ and is given by:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla \phi^{n+1} \tag{3.2}$$

with a pressure update equation such as:

$$p^{n+1/2} = q + \mathbb{L}\phi^{n+1} \tag{3.3}$$

We are going to illustrate the analysis on **Bell, Colella and Glaz**'s inconsistent pressure update first, named here **PmI**:

$$p^{n+1/2} = p^{n-1/2} + \phi^{n+1}$$

followed by using the **corrected pressure update**, named here **PmII**:

$$p^{n+1/2} = p^{n-1/2} + \phi^{n+1} - \frac{\nu \Delta t}{2} \nabla^2 \phi^{n+1},$$

such that $q = p^{n-1/2}$ and $\mathbb{L} = I - \dfrac{\nu \Delta t}{2} \nabla^2$.

3.1 Reference Solution

In this section, we shall derive the solutions for the initial-boundary-value problem for the system given by the partial differential equations Eq. (3.5). We will transform the Stokes equations and solve the system by taking the limit as Δt goes to zero using the relation:

$$z = e^{s \Delta t}, \tag{3.4}$$

which relates the transformed variable for the discrete Laplace transform to that of the continuous Laplace transform. Much of the material and notations in this section follows closely those in Strikerda's book [12] where an introduction to the use of transforms for estimating convergence of solutions for finite difference schemes is given.

In the following sections, we will detail the following steps, that allow us to find the formal order of accuracy of the studied methods:

Find Reference Solutions. Using Fourier and Laplace transforms, we solve the continuous Stokes equations. This is equivalent to using a normal mode ansatz of the form:

$$f(x, y, t) = \hat{f}(x) e^{i\omega y + st}.$$

Find Solutions Discretized in Time. The system of equations here is discrete in time but continuous in space. For this case we perform a Fourier transform in space and a discrete Laplace transform in time and attain the solutions by normal mode analysis.

Temporal Accuracy. By comparing the solutions discretized in time with the "true" solutions, we obtain the temporal accuracy for the different projection methods.

3.1.1 Initial Boundary Value Problem

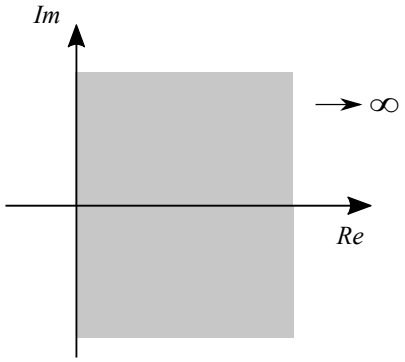


Figure 3.1: Domain of the Initial Boundary Value problem.

We consider the unsteady Stokes problem in the right half plane where $\Omega = [0, \infty) \times [-\pi, \pi]$. This half plane analysis has been proven [10, 15] to be simpler than E and Liu's [11] channel problem, without loss of generality. We now come back to the completely continuous Stokes equations:

$$\begin{cases} \mathbf{u}_t = -\nabla p + \nu \nabla^2 \mathbf{u}, \\ \nabla \cdot \mathbf{u} = 0, \end{cases} \qquad \mathbf{x} \in \Omega, \forall t \geq 0, \tag{3.5}$$

For the existence of a unique solution, we also specify a set of time-varying boundary conditions as follows:

$$u(0, y, t) = \alpha(y, t) \quad \text{and} \quad v(0, y, t) = \beta(y, t). \tag{3.6}$$

Taking the divergence of the Stokes equation and using the divergence-free condition, we arrive at an elliptic equation for the pressure and the resulting system requires an additional boundary condition where the divergence of the velocity equals zero at the boundaries. The need for this extra boundary condition and its eventual equivalence to the original Stokes equations is derived and explained by Henshaw [33]. The new system of equations becomes:

$$\begin{cases} \mathbf{u}_t = -\nabla p + \nu \nabla^2 \mathbf{u}, \\ \nabla^2 p = 0, \end{cases} \qquad \mathbf{x} \in \Omega, \forall t \geq 0, \tag{3.7}$$

with boundary conditions:

$$u(0, y, t) = \alpha, \quad v(0, y, t) = \beta, \quad \nabla \cdot \mathbf{u} = 0, \quad \mathbf{x} \in \partial\Omega. \tag{3.8}$$

We perform a Fourier transform in y and a Laplace transform in time such that:

$$\mathcal{L}\{\partial_t\} \to s,$$

$$\mathcal{F}\{\partial_y\} \to i\omega,$$

where s is the continuous Laplace transform variable and ω is the wave number or spatial frequency of a wave. Transforming the system in Eq. (3.7), we get:

$$\nu(-\partial_x^2 + \mu^2)\hat{u} = -\partial_x \hat{p},$$

$$\nu(-\partial_x^2 + \mu^2)\hat{v} = -i\omega \hat{p},$$

$$(-\partial_x^2 + \omega)\hat{p} = 0,$$

where $\mu^2 = \omega^2 + s/\nu$ and in this case μ is always the positive root.

Solving the ODE for $\hat{p}$

The ordinary differential equation we need to solve is:

$$(-\partial_x^2 + \omega)\hat{p} = 0,$$

with the solution:

$$\hat{p} = Pe^{-|\omega|x}, \tag{3.9}$$

where P is to be determined.

Solving the ODE for $\hat{u}$

The ODE involving $\hat{u}$ is non-homogeneous. We will first solve for the homogeneous solution then use the method of undetermined coefficients to determine the particular solution. The homogeneous solution is:

$$\hat{u}_h = Ue^{-\mu x}. \tag{3.10}$$

We generate the finite family:

$$f_1(x) = -\partial_x \hat{p} = |\omega| Pe^{-|\omega|x} \to \{e^{-|\omega|x}\} \tag{3.11}$$

To find a particular solution u_p corresponding to f_1, we tentatively seek it as a linear combination of terms

in Eq. (3.11):

$$\hat{u}_p(x) = Ae^{-|\omega|x}, \tag{3.12}$$

which we then substitute into the ODE to get:

$$\nu(-A\omega^2 e^{-|\omega|x} + \mu^2 A e^{-|\omega|x}) = |\omega| P e^{-|\omega|x},$$

which implies that:

$$A = \frac{|\omega|}{\nu(\mu^2 - \omega^2)} P.$$

This gives us our particular solution u_p (with the still unknown parameter P) and the final solution for the original ODE:

$$\hat{u} = Ue^{-\mu x} + \frac{|\omega|}{s} P e^{-|\omega|x} \tag{3.13}$$

where we have used the fact that $\mu^2 = \omega^2 + s/\nu$.

Solving the ODE for $\hat{v}$

Using the same method as before, we find that:

$$\hat{v} = Ve^{-\mu x} - \frac{i\omega}{s} P e^{-|\omega|x}. \tag{3.14}$$

Together, Eqs. (3.9), (3.13) and (3.14) form a complete solution to the system of equations (3.7) with the yet undetermined coefficients U, V and P. To find these coefficients we will use the boundary conditions (3.8) in the next section.

3.1.2 Coefficients (U, V, P)

The system we need to solve for the boundary conditions is given by:

$$\begin{bmatrix} 1 & 0 & |\omega|/s \\ 0 & 1 & -i\omega/s \\ -\mu & i\omega & 0 \end{bmatrix} \begin{bmatrix} U \\ V \\ P \end{bmatrix} = \begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \\ 0 \end{bmatrix}$$

This a simple 3×3 system which we can invert using the matrix of cofactors C in the well-known formula:

$$A^{-1} = \frac{1}{\det(A)} C^T,$$

which gives the following coefficients:

$$\begin{cases} U = \dfrac{\nu(\mu+|\omega|)}{s}\left(-|\omega|\hat{\alpha}+i\omega\hat{\beta}\right), \\ V = -\dfrac{i\nu(\mu+|\omega|)\mu}{s}\left(-\dfrac{\omega}{|\omega|}\hat{\alpha}+i\hat{\beta}\right), \\ P = \dfrac{\nu(\mu+|\omega|)}{|\omega|}\left(\mu\hat{\alpha}-i\omega\hat{\beta}\right), \end{cases} \tag{3.15}$$

where ν is a known physical coefficient, μ and ω are given by the normal mode analysis and $(\hat{\alpha},\hat{\beta})$ are known boundary conditions, so we have determined the three coefficients exactly.

3.1.3 Summary

We substitute the coefficients in Eq. (3.15) and get the functional form of the solutions:

$$\begin{cases} \hat{u} = \dfrac{\nu(\mu+|\omega|)}{s}\left(-|\omega|\hat{\alpha}+i\omega\hat{\beta}\right)e^{-\mu x} + \dfrac{\nu(\mu+|\omega|)}{s}\left(\mu\hat{\alpha}-i\omega\hat{\beta}\right)e^{-|\omega|x}, \\ \hat{v} = \dfrac{\nu(\mu+|\omega|)}{s}\left(\dfrac{i\mu\omega}{|\omega|}\hat{\alpha}+\mu\hat{\beta}\right)e^{-\mu x} - \dfrac{\nu(\mu+|\omega|)}{s}\dfrac{\omega}{|\omega|}\left(i\mu\hat{\alpha}+\omega\hat{\beta}\right)e^{-|\omega|x}, \\ \hat{p} = \dfrac{\nu(\mu+|\omega|)}{|\omega|}\left(\mu\hat{\alpha}-i\omega\hat{\beta}\right)e^{-|\omega|x}. \end{cases} \tag{3.16}$$

3.2 Solutions for the Semi-discrete Stokes Equations

3.2.1 Spurious Modes

Eliminating $\mathbf{u}^*$ using Eqs. (3.2), (3.1) gives:

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t} + \nabla\phi^{n+1} + \nabla q = \frac{\nu}{2}\nabla^2(\mathbf{u}^{n+1}+\mathbf{u}^n) + \frac{\nu\Delta t}{2}\nabla^2\nabla\phi^{n+1}. \tag{3.17}$$

We next solve this equation using continuous Fourier (in space) and discrete Laplace (in time) transforms. The Laplace transform of a discrete function v^n on a grid with spacing Δt is defined by:

$$\tilde{v}(z) = \frac{1}{\sqrt{2\pi}}\Delta t \sum_{n=-\infty}^{\infty} z^{-n}v^n, \text{ for } |z|\geq 1, \tag{3.18}$$

while the Fourier transform $\hat{u}(w)$ is defined by:

$$\hat{u}(w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} u(y)\, e^{iwy} \mathrm{d}y, \text{ for } w \in \mathbb{R}. \tag{3.19}$$

We define $\hat{f}(z, w)$ to be both the Fourier transform and discrete Laplace transform of a function $f(t, y)$. We begin our analysis of Eq. (3.17) by taking the Fourier transform in the variable y and the discrete Laplace transform in t. We also let:

$$\hat{q} = z^{n+1} Q(z) \hat{\phi}, \tag{3.20}$$

where $Q(z)$ depends on q and $\mathbb{L}$ in Eq. (3.3). Using the z-transform table with the Fourier Transform, we make the following transformations:

$$\begin{cases} \mathcal{Z}\left\{ \dfrac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} \right\} = \dfrac{1}{\Delta t} \left(\dfrac{z-1}{z} \right) \hat{\mathbf{u}}, \\ \mathcal{Z}\left\{ \mathcal{F}\left\{ \dfrac{\nu}{2} \nabla^2 (\mathbf{u}^{n+1} + \mathbf{u}^n) \right\} \right\} = \dfrac{\nu}{2} (\partial_x^2 - \omega^2) \left(\dfrac{z+1}{z} \right) \hat{\mathbf{u}}, \\ \mathcal{Z}\left\{ \mathcal{F}\left\{ \dfrac{\nu \Delta t}{2} \nabla^2 \nabla \phi^{n+1} \right\} \right\} = \dfrac{\nu \Delta t}{2} (\partial_x^2 - \omega^2) \nabla \hat{\phi}, \\ \mathcal{Z}\left\{ \mathcal{F}\left\{ \nabla \phi^{n+1} \right\} \right\} = \nabla \hat{\phi}, \\ \mathcal{Z}\left\{ \mathcal{F}\left\{ \nabla q \right\} \right\} = Q(z) \nabla \hat{\phi}. \end{cases} \tag{3.21}$$

According to Eq. (3.21), we have:

$$\frac{1}{\Delta t} \left(\frac{z-1}{z} \right) \hat{\mathbf{u}} + \nabla \hat{\phi} + Q(z) \nabla \hat{\phi} = \frac{\nu}{2} (\partial_x^2 - \omega^2) \left(\frac{z+1}{z} \right) \hat{\mathbf{u}} + \frac{\nu \Delta t}{2} (\partial_x^2 - \omega^2) \nabla \hat{\phi}$$

$$\left[\frac{2(z-1)}{\nu \Delta t (z+1)} + (-\partial_x^2 + \omega^2) \right] \hat{\mathbf{u}} = -\frac{z \Delta t}{z+1} \left(-\partial_x^2 + \omega^2 + \frac{2}{\nu \Delta t} \right) \nabla \hat{\phi} - \frac{2 z Q(z)}{\nu (z+1)} \nabla \hat{\phi}$$

To simplify the notation, we introduce the following new variables:

$$\begin{cases} \overline{\mu}^2 = \omega^2 + \rho/\nu, \\ \rho = \dfrac{2(z-1)}{\Delta t (z+1)}, \\ \lambda^2 = \omega^2 + \dfrac{2}{\nu \Delta t}, \end{cases} \tag{3.22}$$

which gives:

$$(-\partial_x^2 + \overline{\mu}^2) \hat{\mathbf{u}} = -\frac{z \Delta t}{z+1} (-\partial_x^2 + \lambda^2) \nabla \hat{\phi} - \frac{2 z Q(z)}{\nu (z+1)} \nabla \hat{\phi}. \tag{3.23}$$

This concludes the proof of Eq. (49) from Brown *et al.*'s Eq. (49). We continue by taking the divergence of Eq. (3.23), whose left-hand side becomes zero as a consequence of the divergence free condition. This leaves:

$$\frac{z\Delta t}{z+1}\left(-\partial_x^2+\lambda^2\right)\nabla\cdot\nabla\hat{\phi}+\frac{2zQ(z)}{\nu(z+1)}\nabla\cdot\nabla\hat{\phi}=0$$

$$\implies\left(-\partial_x^2+\lambda^2+\frac{2Q(z)}{\nu\Delta t}\right)\nabla^2\hat{\phi}=0$$

$$\implies\left(-\partial_x^2+\lambda^2+\frac{2Q(z)}{\nu\Delta t}\right)(-\partial_x^2+\omega^2)\hat{\phi}=0.$$

We now have a proof for Eq. (50) from Brown *et al.*. Furthermore, we can write the solution as $\hat{\phi}=\hat{\phi}_1+\hat{\phi}_2$, where:

$$(-\partial_x^2+\omega^2)\hat{\phi}_1=0, \tag{3.24a}$$

$$\left(-\partial_x^2+\lambda^2+\frac{2Q(z)}{\nu\Delta t}\right)\hat{\phi}_2=0. \tag{3.24b}$$

Brown mentions that $\hat{\phi}_1$ contains the solution we expect to have while $\hat{\phi}_2$ represents a spurious mode in the potential $\hat{\phi}$, which should not appear in the velocity or the pressure. He then mentioned that $\hat{\mathbf{u}}$ does not contain this spurious mode. We will now try to formally prove these statements.

First, we further separate Eq. (3.24b) as follows:

$$Q(z)\hat{\phi}_2=-(-\partial_x^2+\lambda^2)\frac{\nu\Delta t}{2}\hat{\phi}_2,$$

which we can then substitute into Eq. (3.23) to obtain:

$$\begin{aligned}(-\partial_x^2+\overline{\mu}^2)\hat{\mathbf{u}}=&-\frac{z\Delta t}{z+1}\left(-\partial_x^2+\lambda^2\right)\nabla\hat{\phi}_1-\frac{z\Delta t}{z+1}\left(-\partial_x^2+\lambda^2\right)\nabla\hat{\phi}_2\\&-\frac{2zQ(z)}{\nu(z+1)}\nabla\hat{\phi}_1+\frac{2z}{\nu(z+1)}(-\partial_x^2+\lambda^2)\frac{\nu\Delta t}{2}\nabla\hat{\phi}_2.\end{aligned}$$

Solving Eq. (3.24a), gives:

$$\hat{\phi}_1=A_1e^{-|\omega|x}\implies\hat{\phi}=A_1e^{-|\omega|x}+\hat{\phi}_2,$$

which allows us to exactly compute the velocity components:

$$\begin{cases}\hat{u}=Ue^{-\overline{\mu}x}+\dfrac{2z(1+Q(z))}{\rho(z+1)}|\omega|A_1e^{-|\omega|x},\\[2ex]\hat{v}=Ve^{-\overline{\mu}x}-\dfrac{2z(1+Q(z))}{\rho(z+1)}i\omega A_1e^{-|\omega|x}.\end{cases} \tag{3.25}$$

We move on to verify the pressure update equations. From Eqs. (3.3) and (3.20) we have

$$\hat{p} = z^{1/2}(Q(z) + \mathbb{L})\hat{\phi}. \tag{3.26}$$

PmI Analysis

In PmI, $q = p^{n-1/2}$ and $\mathbb{L} = I$. Furthermore, by using Eq. (3.20) and (3.26), we can find that:

$$Q(z) = \frac{1}{z-1}. \tag{3.27}$$

Now that we have an explicit formula for $Q(z)$, we can use it, together with Eq. (3.24b), to find an explicit formula for the spurious mode ϕ_2. Without going into the details, the solution is:

$$\hat{\phi}_2 = A_2 e^{-\tilde{\lambda}x}, \tag{3.28}$$

where:

$$\tilde{\lambda}^2 = \omega^2 + \frac{2z}{\nu\Delta t(z-1)}.$$

Since $\hat{\phi}_2 \neq 0$, we have:

$$\hat{p} = \frac{z^{3/2}}{z-1}\left(A_1 e^{-|\omega|x} + A_2 e^{-\tilde{\lambda}x}\right). \tag{3.29}$$

which *contains the spurious mode.*

PmII Analysis

In PmII, we have:

$$q = p^{n-1/2} \quad \text{and} \quad \mathbb{L} = I - \frac{\nu\Delta t}{2}\nabla^2\hat{\phi},$$

where the $\mathbb{L}$ operator now contains the correction. Since $\hat{p} = z^{1/2}(Q(z) + \mathbb{L})\hat{\phi}$, we will first look to find an explicit formula for $\mathbb{L}\hat{\phi}$:

$$\begin{aligned}\mathbb{L}\hat{\phi} &= \left(I - \frac{\nu\Delta t}{2}\nabla^2\right)\hat{\phi} \\ &= \frac{\nu\Delta t}{2}(-\partial_x^2 + \lambda^2)\hat{\phi},\end{aligned} \tag{3.30}$$

where λ is given in (3.22). By using the previous result and Eqs. (3.20) and (3.26), we get that:

$$Q(z) = \frac{I - \frac{\nu\Delta t}{2}\nabla^2}{z-1}$$

or

$$Q(z)\hat{\phi} = \frac{\nu\Delta t}{2(z-1)}(-\partial_x^2 + \lambda^2)\hat{\phi}. \tag{3.31}$$

At this point, we can use Eq. (3.24b) and the previous result in Eq. (3.31) to get an explicit solution for $\hat{\phi}_2$, which is:

$$\hat{\phi}_2 = A_2 e^{-\lambda x}. \tag{3.32}$$

From Eq. (3.26), we have that:

$$\begin{aligned}
\hat{p} &= z^{1/2}\left(Q(z) + \mathbb{L}\right)\hat{\phi} \\
&= z^{1/2}\left(\frac{\mathbb{L}\hat{\phi}}{z-1} + \mathbb{L}\hat{\phi}\right) \\
&= \frac{z^{3/2}}{z-1}\frac{\nu\Delta t}{2}(-\omega^2 A_1 e^{-|\omega|x} + \lambda^2 A_1 e^{-|\omega|x} \cancel{-\lambda^2 A_2 e^{-\lambda x}} + \cancel{\lambda^2 A_2 e^{-\lambda x}}),
\end{aligned}$$

where we can clearly see, the improved pressure update formula in PmII has its spurious mode $\hat{\phi}_2$ canceled. This gives a final solution:

$$\hat{p} = \frac{z^{3/2}}{z-1} A_1 e^{-|\omega|x}. \tag{3.33}$$

3.2.2 Application of Boundary Conditions

We'll now follow closely the proofs for Brown *et al.*'s time-varying boundary conditions and include a detailed derivation of each and every point made in his paper [15]. As stated in (3.8), the boundary conditions are:

$$u^* = \alpha, \quad \phi_x = 0, \quad v^* = \beta + \Delta t\tilde{\phi}_y, \quad \text{and} \quad u_x + v_y = 0,$$

where $\tilde{\phi}$ is an approximation of ϕ^{n+1} with varying orders of accuracy. To approximate ϕ^{n+1}, we use an expansion in Δt:

$$\tilde{\phi} = \tilde{\phi}_0 + \Delta t\tilde{\phi}_1 + \Delta t^2\tilde{\phi}_2 + \cdots,$$

where the terms up to second-order are:

Zeroth Order. The zeroth order approximation is trivially:

$$\tilde{\phi} = 0.$$

First Order. To first-order, we have:

$$\tilde{\phi} = \phi^n,$$

because $\phi^{n+1} - \phi^n = \mathcal{O}(\Delta t)$, so $\tilde{\phi} \approx \phi^{n+1} \approx \phi^n$ to first-order.

Second Order . To second-order, we have:

$$\tilde{\phi} = 2\phi^n - \phi^{n-1},$$

from a second-order approximation of $\partial_t \phi$.

We then perform Fourier and discrete Laplace transform on the boundary conditions,

$$\hat{u} = \hat{\alpha}, \quad \hat{\phi}_x = 0, \quad \hat{v} = \hat{\beta} - i\omega\Delta t B(z)\hat{\phi} \quad \text{and} \quad \hat{u}_x + i\omega\hat{v} = 0, \tag{3.34}$$

where $B(z)$ is a function that depends on choice of the approximation $\tilde{\phi}$ and can be determined by applying a Laplace transform on the consecutive approximations as follows:

Zeroth order.

$$\mathcal{Z}\{\mathcal{F}\{\phi^{n+1}\}\} = \hat{\phi} \implies B(z) = 1.$$

First order.

$$\mathcal{Z}\{\mathcal{F}\{\phi^{n+1} - \phi^n\}\} = \left(\frac{z-1}{z}\right)\hat{\phi} \implies B(z) = \frac{z-1}{z}.$$

Second order.

$$\mathcal{Z}\{\mathcal{F}\{\phi^{n+1} - 2\phi^n + \phi^{n-1}\}\} = \frac{(z-1)^2}{z^2}\hat{\phi} \implies B(z) = \frac{(z-1)^2}{z^2}.$$

3.2.3 Coefficients (U, V, A_1, A_2)

In this section, we would like to use the appropriate boundary conditions obtained in the previous section in an attempt to find the coefficients (U, V, A_1) for the general solutions in (3.25) and A_2 from:

$$\hat{\phi} = A_1 e^{-|\omega|x} + A_2 e^{-\gamma x}, \tag{3.35}$$

where

$$\gamma^2 = \omega^2 + \frac{2}{\nu\Delta t}F(z).$$

and $F(z)$ depends on whether we are looking at the PmI or the PmII pressure update scheme.

Finding A_1

Using Eq. (3.25) and the above boundary conditions,

$$\hat{u} = \hat{\alpha} \qquad \text{and} \qquad \hat{u} = Ue^{-\overline{\mu}x} + R(z)\frac{|\omega|}{\rho}A_1 e^{-|\omega|x},$$

we get:

$$\hat{\alpha} = U + R(z)\frac{|\omega|}{\rho}A_1. \tag{3.36}$$

We will now do a series of substitution and try to isolate a single coefficient with which we can express the others. First, from the boundary condition $\hat{\phi}_x = 0$, we get the relationship $A_2 = -|\omega|A_1/\gamma$. We substitute this into $\hat{\phi} = A_1 e^{-|\omega|x} + A_2 e^{-\gamma x}$ such that $\hat{\phi}$ is written only in terms of A_1. We then substitute this new $\hat{\phi}$ into the boundary condition $\hat{v} = \hat{\beta} - i\omega\Delta t B(z)\hat{\phi}$ such that $\hat{v}$ is also expressed only in terms of one coefficient, namely A_1. Following that, we substitute this new $\hat{v}$ into the third boundary condition $\hat{u}_x + i\omega\hat{v} = 0$. Now that we have a tentative plan to derive the coefficient A_1, we will look in detail at each step.

First, we substitute $A_2 = -|\omega|A_1/\gamma$ into Eq. (3.35):

$$\hat{\phi} = A_1 - \frac{|\omega|A_1}{\gamma}. \tag{3.37a}$$

Then, we substitute Eq. (3.37a) into $\hat{v} = \hat{\beta} - i\omega\Delta t B(z)\hat{\phi}$:

$$\hat{v} = \hat{\beta} - i\omega\Delta t B(z)\left(A_1 - \frac{|\omega|A_1}{\gamma}\right). \tag{3.37b}$$

Finally, we substitute Eq. (3.37b) into $\hat{u}_x + i\omega\hat{v} = 0$:

$$\overbrace{-\overline{\mu}U - |\omega|R(z)\frac{|\omega|}{\rho}A_1}^{\hat{u}_x} + i\omega\overbrace{\left(\hat{\beta} - i\omega\Delta t B(z)\left(A_1 - \frac{|\omega|A_1}{\gamma}\right)\right)}^{\hat{v}} = 0$$

$$\overline{\mu}U + R(z)\frac{\omega^2}{\rho}A_1 - \omega^2\Delta t B(z)\left(A_1 - \frac{|\omega|A_1}{\gamma}\right) = i\omega\hat{\beta}$$

$$\overline{\mu}U + \omega^2 A_1\left[\frac{R(z)}{\rho} - \Delta t B(z)\left(1 - \frac{|\omega|}{\gamma}\right)\right] = i\omega\hat{\beta}$$

which, by substituting $\gamma^2 = \omega^2 + \dfrac{2}{\nu\Delta t}F(z)$, and $\Delta t = \dfrac{2F(z)}{\nu(\gamma^2 - \omega^2)}$, gives:

$$\overline{\mu}U + \omega^2 A_1\left[\frac{R(z)}{\rho} - \left(\frac{2F(z)B(z)}{\nu(\gamma^2 - \omega^2)}\right)\left(\frac{\gamma - |\omega|}{\gamma}\right)\right] = i\omega\hat{\beta}$$

$$\overline{\mu}U + \omega^2 A_1\left[\frac{R(z)}{\rho} - \frac{2F(z)B(z)}{\nu\gamma(\gamma + |\omega|)}\right] = i\omega\hat{\beta} \tag{3.37c}$$

Recall that in Eq. (3.36), $\hat{\alpha} = U + R(z)\frac{|\omega|}{\rho}A_1$. We substitute $U = \hat{\alpha} - R(z)\frac{|\omega|}{\rho}A_1$ into Eq. (3.37c) to get:

$$\overline{\mu}\hat{\alpha} - \overline{\mu}R(z)\frac{|\omega|}{\rho}A_1 + A_1\left[\frac{\omega^2 R(z)}{\rho} - \frac{2\omega^2 F(z)B(z)}{\nu\gamma(\gamma+|\omega|)}\right] = i\omega\hat{\beta}$$

$$A_1\left[1 + \frac{2\rho|\omega|F(z)B(z)}{R(z)(\overline{\mu}-|\omega|)\nu\gamma(\gamma+|\omega|)}\right] = \frac{\rho(\overline{\mu}\hat{\alpha} - i\omega\hat{\beta})}{R(z)|\omega|(\overline{\mu}-|\omega|)} \tag{3.38}$$

Recall that $\overline{\mu}^2 = \omega^2 + \rho/\nu$. Substitute $\rho = \nu(\overline{\mu}^2 - \omega^2)$ to get:

$$A_1\left[1 + \frac{2|\omega|\nu(\overline{\mu}^2-\omega^2)F(z)B(z)}{R(z)(\overline{\mu}-|\omega|)\nu\gamma(\gamma+|\omega|)}\right] = \frac{\nu(\overline{\mu}^2-\omega^2)(\overline{\mu}\hat{\alpha} - i\omega\hat{\beta})}{R(z)|\omega|(\overline{\mu}-|\omega|)}$$

$$A_1 = \frac{\nu(\overline{\mu}+|\omega|)(\overline{\mu}\hat{\alpha} - i\omega\hat{\beta})}{R(z)|\omega|}\left[1 + C\frac{F(z)B(z)}{R(z)}\right]^{-1} \tag{3.39}$$

where $C = \dfrac{2|\omega|(\overline{\mu}+|\omega|)}{\gamma(\gamma+|\omega|)}$. So, Eq. (3.39) gives us an explicit formula for A_1.

Finding U

From Eq. (3.36), $\hat{\alpha} = U + R(z)\frac{|\omega|}{\rho}A_1$. We see that $U = \hat{\alpha} - R(z)\frac{|\omega|}{\rho}A_1$. We'll substitute A_1 into this equation to find coefficient U:

$$\begin{aligned} U &= \hat{\alpha} - R(z)\frac{|\omega|}{\rho}A_1 \\ &= \left[\frac{\rho\hat{\alpha} - \nu(\overline{\mu}+|\omega|)(\overline{\mu}\hat{\alpha} - i\omega\hat{\beta})}{\rho} + \hat{\alpha}C\frac{F(z)B(z)}{R(z)}\right]\left[1 + C\frac{F(z)B(z)}{R(z)}\right]^{-1} \end{aligned} \tag{3.40}$$

Again recall that $\overline{\mu}^2 = \omega^2 + \rho/\nu$. Substitute $\rho = \nu(\overline{\mu}^2 - \omega^2)$ only to the numerator.

$$= \left[\frac{\nu(\overline{\mu}^2-\omega^2)\hat{\alpha} - \nu(\overline{\mu}+|\omega|)(\overline{\mu}\hat{\alpha} - i\omega\hat{\beta})}{\rho} + \hat{\alpha}C\frac{F(z)B(z)}{R(z)}\right]\left[1 + C\frac{F(z)B(z)}{R(z)}\right]^{-1}$$

$$U = \left[\frac{\nu(\overline{\mu}+|\omega|)(i\omega\hat{\beta} - |\omega|\hat{\alpha})}{\rho} + \hat{\alpha}C\frac{F(z)B(z)}{R(z)}\right]\left[1 + C\frac{F(z)B(z)}{R(z)}\right]^{-1}. \tag{3.41}$$

Finding V

From the transformed bc in Eq. (3.34) where $\hat{u}_x + i\omega\hat{v} = 0$, and substitute the solutions for $\hat{u}$ and $\hat{v}$ in Eq. (3.25). We get:

$$-\overline{\mu}Ue^{-\overline{\mu}x} - |\omega|R(z)\frac{|\omega|}{\rho}A_1e^{-|\omega|x} + i\omega\left(Ve^{-\overline{\mu}x} - R(z)\frac{i\omega}{\rho}A_1e^{-|\omega|x}\right) = 0$$

$$\implies V = -\frac{i\overline{\mu}}{\omega}U \tag{3.42}$$

where U is already found in Eq. (3.41) and for simplicity, we're not going to state it in full.

Finding A_2

We start from Eq. (3.35) and use bc in Eq. (3.34) where $\hat{\phi}_x = 0$,

$$\hat{\phi} = A_1e^{-|\omega|x} + A_2e^{-\gamma x}$$

$$\hat{\phi}_x = -|\omega|A_1 - \gamma A_2 \equiv 0$$

$$A_2 = \frac{-|\omega|}{\gamma}A_1 \tag{3.43}$$

where A_1 is already found in Eq. (3.39) and for simplicity, we're also not going to state it in full.

3.2.4 Summary

To summarize, we restate the solutions in Eqs. (3.25) and (3.35) for the Stokes equations discretized in time.

$$\begin{cases} \hat{u} = Ue^{-\overline{\mu}x} + R(z)\dfrac{|\omega|}{\rho}A_1e^{-|\omega|x}, \\ \hat{v} = Ve^{-\overline{\mu}x} - R(z)\dfrac{i\omega}{\rho}A_1e^{-|\omega|x}, \\ \hat{\phi} = A_1e^{-|\omega|x} + A_2e^{-\gamma x}, \end{cases} \tag{3.44}$$

where

$$R(z) = \frac{2z(1+Q(z))}{(z+1)} \quad \text{and} \quad \gamma^2 = \omega^2 + \frac{2}{\nu\Delta t}F(z).$$

The coefficients (U, V, A_1 and A_2) are in Eqs. (3.41), (3.42), (3.39) and (3.43) respectively. They are restated below:

$$\begin{cases} U = \left[\dfrac{\nu(\overline{\mu}+|\omega|)(i\omega\hat{\beta}-|\omega|\hat{\alpha})}{\rho} + \hat{\alpha}C\dfrac{F(z)B(z)}{R(z)}\right]\left[1+C\dfrac{F(z)B(z)}{R(z)}\right]^{-1} \\ V = -\dfrac{i\overline{\mu}}{\omega}U \\ A_1 = \dfrac{\nu(\overline{\mu}+|\omega|)(\overline{\mu}\hat{\alpha}-i\omega\hat{\beta})}{R(z)|\omega|}\left[1+C\dfrac{F(z)B(z)}{R(z)}\right]^{-1} \\ A_2 = \dfrac{-|\omega|}{\gamma}A_1 \end{cases} \tag{3.45}$$

where

$$C = \frac{2|\omega|(\overline{\mu}+|\omega|)}{\gamma(\gamma+|\omega|)}.$$

Note that the choice of q and $\mathbb{L}$ in the pressure update formula determines $F(z)$ and $R(z)$, while $B(z)$ is determined by the accuracy of boundary conditions used.

3.3 Order of Accuracy

We first compare the coefficient U in Eq. (3.41) with the reference solution Eq. (3.15). We noticed that the first term differs in terms of denominator:

$$\text{Reference } U: \quad \frac{\nu(\mu+|\omega|)}{s}\left(-|\omega|\hat{\alpha}+i\omega\hat{\beta}\right). \qquad \text{Discretized } U: \quad \frac{\nu(\overline{\mu}+|\omega|)(i\omega\hat{\beta}-|\omega|\hat{\alpha})}{\rho}.$$

We need to determine the order of accuracy of ρ.

$$\rho = \frac{1}{\Delta t}\frac{2(z-1)}{(z+1)}.$$

Recall that in Eq. (3.4), $z = e^{s\Delta t}$, so:

$$\rho = \frac{1}{\Delta t}\frac{2(e^{s\Delta t}-1)}{(e^{s\Delta t}+1)},$$

which, using a MacLaurin series expansion for $e^{s\Delta t}$, gives:

$$\begin{aligned} \rho &= \frac{1}{\Delta t}\left(s\Delta t - \frac{s^3\Delta t^3}{12} + O(s^5\Delta t^5)\right) \\ \implies \rho &= s + \mathcal{O}(s^3\Delta t^2). \end{aligned}$$

Therefore, this first term is second-order as the first error term in denominator is $\mathcal{O}(s^3\Delta t^2)$ compared to reference solution. One can also prove that since $\rho = s + \mathcal{O}(s^3\Delta t^2)$, this leads to $\overline{\mu} = \mu + \mathcal{O}(s^2\Delta t^2)$. In any case, going to the second term

$$C\frac{F(z)B(z)}{R(z)}$$

gives that this particular term has to be $\mathcal{O}(\Delta t^2)$. However, the denominator of C is $\gamma(\gamma + |\omega|)$, which is not easy to analyze with respect to its order. We take advantage of the fact that $\gamma(\gamma+|\omega|) \geq \gamma^2 = \omega^2 + \frac{2}{\nu\Delta t}F(z)$, and analyze γ^2 instead. Therefore,

$$\begin{aligned} C\frac{F(z)B(z)}{R(z)} &\leq \frac{2|\omega|(\overline{\mu}+|\omega|)}{\gamma^2}\frac{F(z)B(z)}{R(z)} \\ &\leq \frac{2|\omega|(\overline{\mu}+|\omega|)}{\frac{\omega^2\nu\Delta t + 2F(z)}{\nu\Delta t}}\frac{F(z)B(z)}{R(z)} \\ &\leq 2|\omega|\nu(\overline{\mu}+|\omega|)\frac{\Delta t F(z)B(z)}{R(z)[2F(z)+\omega^2\nu\Delta t]}. \end{aligned}$$

Therefore it suffices to show that:

$$\frac{F(z)B(z)}{R(z)[2F(z)+\omega^2\nu\Delta t]} \sim \mathcal{O}(\Delta t).$$

3.3.1 PmI Analysis

We previously derived $Q(z) = 1/(z-1)$ in Eq. (3.27) for PmI. Therefore,

$$R(z) = \frac{2z(1+Q(z))}{(z+1)} = \mathcal{O}(\Delta t^{-1}). \qquad \text{(by MacLaurin series expansion)}$$

We also recall that from Eq. (3.28) that $\gamma^2 = \omega^2 + \frac{2}{\nu\Delta t}\frac{z}{(z-1)}$, such that:

$$F(z) = \frac{z}{z-1} = \mathcal{O}(\Delta t^{-1}). \qquad \text{(by MacLaurin series expansion)}$$

and hence:

$$\frac{B(z)F(z)}{R(z)[2F(z)+\omega^2\nu\Delta t]} \sim B(z)\mathcal{O}(\Delta t).$$

Therefore, $B(z) = \mathcal{O}(1)$ which means that the zeroth order boundary condition $v^* = \beta$ (corresponding to

$\tilde{\phi} = 0$) can be used. We recall Eq. (3.29):

$$\begin{aligned}\hat{p} &= \frac{z^{3/2}}{z-1}(\hat{\phi}_1 + \hat{\phi}_2) \\ &= \frac{z^{3/2}}{z-1}\hat{\phi}_1 + \frac{z^{3/2}}{z-1}\hat{\phi}_2 \\ &\sim \mathcal{O}\left(\frac{z^{3/2}}{z-1}A_1 + \frac{z^{3/2}}{z-1}A_2\right)\end{aligned}$$

Order of Accuracy for the Pressure

Lets first prove that $\frac{z^{3/2}}{z-1}A_1 \sim \mathcal{O}(\Delta t^2)$. Since we've already proven earlier that $U = \mathcal{O}(\Delta t^2)$, from Eq. (3.36) we have:

$$\begin{aligned}\hat{\alpha} = \underbrace{U}_{\mathcal{O}(\Delta t^2)} + \underbrace{R(z)}_{\mathcal{O}((s\Delta t)^{-1})} \frac{|\omega|}{\rho}A_1 \\ \implies A_1 = \mathcal{O}(s^2\Delta t^3) \\ \implies \frac{z^{3/2}}{z-1}A_1 \sim \mathcal{O}(\Delta t^2).\end{aligned} \tag{3.46}$$

Let's prove that $\frac{z^{3/2}}{z-1}A_2 \sim \mathcal{O}(\nu\Delta t)$. Starting from $\gamma^2 = \omega^2 + \frac{2}{\nu\Delta t}F(z)$ and since $F(z) = \mathcal{O}((s\Delta t)^{-1})$, we have:

$$\begin{aligned}\gamma^2 &= \omega^2 + \frac{2}{\nu\Delta t}F(z) \\ &\vdots \\ \gamma &= \underbrace{\lambda + \mathcal{O}(\nu\Delta t)}_{\tilde{\lambda}}.\end{aligned}$$

We notice that $A_2 = -|\omega|A_1/\gamma$ and hence $A_2 \sim \mathcal{O}(\gamma) + \mathcal{O}(A_1)$. Since γ has a lower order, the order of accuracy of pressure's spurious mode has a leading error term of order γ.

$$\frac{z^{3/2}}{z-1}A_2 \sim \mathcal{O}(\nu\Delta t) \tag{3.47}$$

So the final order of accuracy for the pressure is:

$$\hat{p} = \mathcal{O}(\nu\Delta t). \tag{3.48}$$

Pressure is only expected to be first-order in time with PmI due to the presence of spurious mode A_2 in $\hat{p}$.

Order of Accuracy for the Velocity

Comparing the coefficients of U between the discretized system and the reference solution we see that for either PmI or PmII, the velocity are always $\mathcal{O}(\Delta t^2)$. This is further confirmed by an earlier analysis that $\hat{\mathbf{u}}$ do not contain the spurious modes in pressure regardless of pressure update formula. So the formal order of the velocity is:

$$\hat{\mathbf{u}} = \mathcal{O}(\Delta t^2). \tag{3.49}$$

Order of Accuracy for the Boundary Conditions

Since $B(z) \sim \mathcal{O}(1)$, zeroth order accurate boundary conditions are permitted. This means that we are allowed to use:

$$v^* = \beta \implies \tilde{\phi} = 0. \tag{3.50}$$

3.3.2 PmII Analysis

Previously, in PmII's analysis of its pressure modes, we observed the following in Eq. (3.31):

$$Q(z)\hat{\phi} = \frac{\mathbb{L}\hat{\phi}}{z-1},$$
$$Q(z) = \frac{\mathbb{L}}{z-1},$$

noting that $\mathbb{L}$ is not a function of the discrete Laplace variable, z

$$Q(z) = \frac{1}{z-1}$$

which is the same as PmI. This leads to the same $R(z)$:

$$\begin{aligned} R(z) &= \frac{2z(1+Q(z))}{z+1} \\ &= \frac{2z^2}{z^2-1} \sim \mathcal{O}((s\Delta t)^{-1}). \end{aligned}$$

We recall that in PmII $\gamma = \lambda$ such that $F(z) = 1$, so:

$$\frac{B(z)F(z)}{R(z)[2F(z) + \omega^2\nu\Delta t]} \sim B(z)\mathcal{O}(\Delta t).$$

Again, $B(z) = \mathcal{O}(1)$ which means that the zeroth order boundary condition $v^* = \beta$ can be used. In this case,

both the velocity and the pressure are expected to converge to second-order in time. Zeroth order boundary conditions are also permitted.

3.3.3 Comments on Higher Order Boundary Conditions

It might be curious to the reader why are boundary conditions with higher accuracy introduced in the scheme. This is because it can be proven [15] that Kim and Moin's [13] fractional step method [13] requires at least first-order accurate boundary conditions (i.e. $v^* = \beta + \Delta t \tilde{\phi}_y$, where $\tilde{\phi} = \phi^n$) in order for the pressure to be second-order in time. Many others [10, 11] have proven that using the boundary conditions proposed by Kim and Moin leads to first-order accurate pressure updating. Brown *et al.* [15] saw that the pressure recovery equation in Kim and Moin's paper should be thought of as the time-centered pressure in Eq. (1.16) in order for global second-order accuracy to be achieved. Even though the analysis is not stated in this report, we did follow the necessary steps that led up to Brown's conclusions.

Chapter 4

Numerical Tests

We solve the Navier-Stokes equations using Crank-Nicolson for both the convective and the viscous terms, with Bell, Colella and Glaz's [8] pressure update scheme:

$$M\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t}+N\mathbf{u}^{n+1}+Gp^{n+1/2}=\frac{\nu}{2}L(\mathbf{u}^{n+1}+\mathbf{u}^n)+\boldsymbol{mbc}, \tag{4.1}$$

$$D\mathbf{u}^{n+1}=cbc, \tag{4.2}$$

where we refer to Beam-Warming [17] and Eq. (4.17) for the linearization of the convective term $N\mathbf{u}^{n+1}$. In this case, the fractional step method can be written as:

Step 1: Solve for the intermediate velocity $\mathbf{u}^*$:

$$A\mathbf{u}^*=\boldsymbol{r}+\boldsymbol{mbc}, \tag{4.3}$$

where

$$A=\frac{1}{\Delta t}M+N-\frac{\nu}{2}L, \tag{4.4}$$

$$\boldsymbol{r}=\left[\frac{1}{\Delta t}M+\frac{\nu}{2}L\right]\mathbf{u}^n-Gp^{n-1/2}. \tag{4.5}$$

Step 2: Perform the projection to obtain the divergence-free velocity $\mathbf{u}^{n+1}$:

$$\mathbf{u}^{n+1}=\mathbf{u}^*-\Delta tM^{-1}G\phi^{n+1}, \tag{4.6}$$

where

$$\Delta tDM^{-1}G\phi^{n+1}=D\mathbf{u}^*+cbc. \tag{4.7}$$

Step 3: Pressure update:

$$p^{n+1/2}=p^{n-1/2}+\phi^{n+1}. \tag{4.8}$$

Since Step 1 solves for the intermediate velocity which lies on the face centers, it is solved on the staggered grid. Step 2, which is the Poisson equation, solves for the auxiliary pressure variable ϕ^{n+1}, which lies on the

cell centers. Therefore solving for Eq. (4.7) involves inverting a cell-centered Laplace operator.

The duration of this research project did not allow us to stretch our numerical tests to the extent where we could test the results of our normal mode analysis in a more rigorous manner that Brown *et al.* [15] did. Nevertheless, a Navier-Stokes solver was implemented with the first-order pressure update scheme proposed by Bell, Colella and Glaz [8]. As will be seen, our numerical results shows Bell, Colella and Glaz's pressure update scheme to be global second-order. This is discordant with the results obtained in the normal mode analysis which predicted this pressure update scheme to be first-order in time. The problem lies in the fact that this simple numerical test conducted in a periodic box is unable to show the "true colors" of this first-order pressure update scheme. We chose this numerical test because it is a classic test conducted by Perot [4]. K.Kim *et al.* [7] also used the same scheme and numerical tests and achieved results of global second-order accuracy in a similar periodic test. Essentially, this test problem was not able to reveal the spurious mode in the pressure update scheme (revealed in Section 3). Brown *et al.* was able to come up with test problems that reveals the problematic nature of Bell, Colella and Glaz's inconsistent pressure update scheme and they matched well with the results of the normal mode analysis in Section 3.

4.1 Discrete Operators

We use Morinishi's [2] operators and we propose to include a mesh element's surface areas and volume directly so that non-uniform meshes can be handled seamlessly.

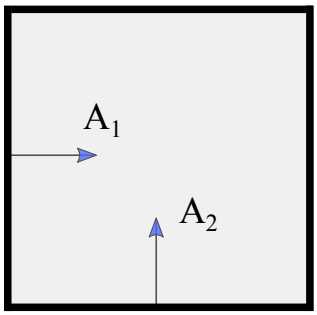


Figure 4.1: Definition of area vectors

We will generally refer the reader to [2] for an introduction to the notation used in this chapter. The first operator we will introduce is the zeroth order geometric surface law, written as:

$$(\mathrm{GSL})_i = \frac{\delta A_i}{\delta \xi_i}, \tag{4.9}$$

where A_i denotes the area of a face perpendicular to the i-th axis and ξ_i denote the logical coordinates. Underlying this formula is the choice we made to have a unitary logical mesh spacing $\Delta\xi_i = 1$ between two mesh points. We also note that the geometric surface law (or discrete divergence theorem) has to be satisfied in every cell:

$$\int_{\partial\Omega} \hat{\mathrm{n}}_i \, \mathrm{dS} \;\equiv\; \sum_i \frac{\delta A_i}{\delta \xi_i} = 0 \implies (\mathrm{GSL})_i = 0. \tag{4.10}$$

Discrete Continuity Operator (On a Regular Grid)

Using Eq. (4.10), we define our continuity operator as follows:

$$(\mathrm{Cont.})_i = \sum_i \frac{\delta A_i u_i}{\delta \xi_i} = 0. \tag{4.11}$$

We also note that Eqs. (4.10) and (4.11) remain true in the context of non-uniform cells. An exercise to test the conservation properties of our viscous and convective operators are detailed in later sections 4.1.1 and 4.1.2, respectively. Higher order conservation properties in the staggered grid system have already been proven by Morinishi *et al.* [2].

4.1.1 Viscous Term

The symmetric gradient tensor of a vector is defined as:

$$e_{ij} = \frac{1}{2V_{ij}} \left(\frac{\delta \overline{A_i}^{\,i^j} u_j}{\delta \xi_i} + \frac{\delta \overline{A_j}^{\,i^j} u_j}{\delta \xi_j} \right),$$

where the notation $\overline{A}^{\,i}$ denotes an interpolation along the i-th axis of the given quantity. An equivalent form is:

$$e_{ij} = \frac{1}{2V_{ij}} \left(\overline{A_i}^{\,j} \frac{\delta u_j}{\delta \xi_i} + \overline{A_j}^{\,i} \frac{\delta u_i}{\delta \xi_j} + \overline{u_j \overline{(\mathrm{GSL})_i}^{\,j}}^{\,i} + \overline{u_i \overline{(\mathrm{GSL})_j}^{\,i}}^{\,j} \right), \tag{4.12}$$

which can be proved using the geometric surface law defined before. We also define the divergence of a discrete rank 2 tensor as:

$$[\mathrm{viscous}]_i \equiv \sum_j \left(\frac{\delta \overline{A_j}^{\,i} e_{ij}}{\delta \xi_j} \right). \tag{4.13}$$

Proof of Conservation Properties

We assume the fluid to be Newtonian with a stress tensor defined by $\tau_{ij} = 2\mu e_{ij} + \lambda(\overset{0}{\nabla\cdot\mathbf{u}})\delta_{ij}$. We then take the dot product between the stress tensor and u_j to get:

$$\begin{aligned}\overline{u_j \frac{\delta \overline{A_i}^j \tau_{ij}}{\delta\xi_i}}^j &= \overline{u_j \frac{\delta}{\delta\xi_i}\left(2\mu \frac{\overline{A_i}^j}{2V_{ij}}\left(\overline{A_i}^j \frac{\delta u_j}{\delta\xi_i} + \overline{A_j}^i \frac{\delta u_i}{\delta\xi_j}\right)\right)}^j \\ &= \underbrace{\frac{\delta}{\delta\xi_i}\overline{\left(2\mu\overline{u_j}^i \overline{A_i}^j e_{ij}\right)}^j}_{\text{conservative term}} - \underbrace{\overline{\frac{\mu}{2V_{ij}}\left(\overline{A_i}^j \frac{\delta u_j}{\delta\xi_i} + \overline{A_j}^i \frac{\delta u_i}{\delta\xi_j}\right)^2}^{ij}}_{\text{signed negative term}}. \end{aligned} \tag{4.14}$$

We can see that we have obtained a conservative term (in the sense that it represents a divergence of a certain quantity) and a signed (negative) term for first portion of τ_{ij}. This negative (signed) term in the discrete kinetic energy budget is a sink term which is beneficial for stability.

4.1.2 Convective Term

We derive the linearization of convective term as presented by Beam and Warming [17]. Such a linearization technique results in second-order temporal accuracy as shown below. The linearization is obtained from a Taylor series expansion as follows:

$$\begin{aligned} u_i^{n+1}u_j^{n+1} &= u_i^n u_j^n + \Delta t \frac{\partial}{\partial t}(u_i^n u_j^n) + O(\Delta t^2) \\ &= u_i^n u_j^n + \Delta t\left(u_j^n \frac{\partial u_i^n}{\partial t} + u_i^n \frac{\partial u_j^n}{\partial t}\right) + O(\Delta t^2) \\ &= u_i^n u_j^n + \Delta t\left(u_j^n \frac{u_i^{n+1} - u_i^n}{\Delta t} + u_i^n \frac{u_j^{n+1} - u_j^n}{\Delta t} + O(\Delta t)\right) + O(\Delta t^2) \\ &= u_i^n u_j^n + u_j^n u_i^{n+1} - u_i^n u_j^n + u_i^n u_j^{n+1} - u_i^n u_j^n + O(\Delta t^2) \\ &= u_i^{n+1}u_j^n + u_i^n u_j^{n+1} - u_i^n u_j^n + O(\Delta t^2). \end{aligned} \tag{4.15}$$

Next, we apply the Crank-Nicolson treatment of convective terms as presented by Kim *et al.* [7] and linearize

it using Eq. (4.15):

$$\begin{aligned}\frac{1}{2}H(\mathbf{u}^{n+1})+\frac{1}{2}H(\mathbf{u}^{n}) &= \frac{1}{2}\frac{\partial}{\partial x_j}(u_i^{n+1}u_j^{n+1})+\frac{1}{2}\frac{\partial}{\partial x_j}(u_i^{n}u_j^{n})\\ &= \frac{1}{2}\frac{\partial}{\partial x_j}(u_i^{n+1}u_j^{n}+u_i^{n}u_j^{n+1}-u_i^{n}u_j^{n})+\frac{1}{2}\frac{\partial}{\partial x_j}(u_i^{n}u_j^{n})\\ &= \frac{1}{2}\frac{\partial}{\partial x_j}(u_i^{n+1}u_j^{n}+u_i^{n}u_j^{n+1}). \end{aligned} \tag{4.16}$$

We express Eq. (4.16) as a linear convective operator N such that:

$$N\mathbf{u}^{n+1}=\frac{1}{2}H(\mathbf{u}^{n+1})+\frac{1}{2}H(\mathbf{u}^{n}). \tag{4.17}$$

In the numerical tests presented in later sections, we use a divergence form [2] to discretize Eq. (4.16) where $\frac{\partial}{\partial x_j}(u_iu_j)\simeq\sum_j\frac{\delta\overline{A_ju_j}^{i}\overline{u_i}^{j}}{\delta\xi_j}$. To illustrate this better, the N operator that results from Eq. (4.17) for a 3-D case is as follows:

$$\begin{bmatrix} \frac{1}{2}\left(2\frac{\delta\overline{A_1u^n}^{1}}{\delta\xi_1}+\frac{\delta\overline{A_2v^n}^{1}}{\delta\xi_2}+\frac{\delta\overline{A_3w^n}^{1}}{\delta\xi_3}\right) & \frac{1}{2}\frac{\delta\overline{A_2}^{1}\overline{u^n}^{2}}{\delta\xi_2} & \frac{1}{2}\frac{\delta\overline{A_3}^{1}\overline{u^n}^{3}}{\delta\delta\xi_3}\\ \frac{1}{2}\frac{\delta\overline{A_1}^{2}\overline{v^n}^{1}}{\delta\xi_1} & \frac{1}{2}\left(\frac{\delta\overline{A_1u^n}^{2}}{\delta\xi_1}+2\frac{\delta\overline{A_2v^n}^{2}}{\delta\xi_2}+\frac{\delta\overline{A_3w^n}^{2}}{\delta\xi_3}\right) & \frac{1}{2}\frac{\delta\overline{A_3}^{2}\overline{v^n}^{3}}{\delta\xi_3}\\ \frac{1}{2}\frac{\delta\overline{A_1}^{3}\overline{w^n}^{1}}{\delta\xi_1} & \frac{1}{2}\frac{\delta\overline{A_2}^{3}\overline{w^n}^{2}}{\delta\xi_2} & \frac{1}{2}\left(\frac{\delta\overline{A_1u^n}^{3}}{\delta\xi_1}+\frac{\delta\overline{A_2v^n}^{3}}{\delta\xi_2}+2\frac{\delta\overline{A_3w^n}^{3}}{\delta\xi_3}\right) \end{bmatrix}$$

Proof of Conservation Properties

We take the dot product of u_i with the i^{th} component of convective discretization and use Morinishi's identity in Eq. (42) from his paper [2], to get:

$$\begin{aligned} u_i\sum_j\frac{\delta\overline{u_jA_j}^{i}\overline{u_i}^{j}}{\delta\xi_j} &= \sum_j\left[\frac{1}{2}\frac{\delta\overline{u_jA_j}^{i}\cdot\widetilde{u_iu_i}^{j}}{\delta\xi_j}+\frac{u_iu_i}{2}\frac{\delta_1\overline{u_jA_j}^{i}}{\delta\xi_j}\right]\\ &= \underbrace{\sum_j\left[\frac{\delta\overline{u_jA_j}^{i}\cdot\widetilde{u_iu_i}^{j}/2}{\delta\xi_j}\right]}_{\text{conservative term}}+\frac{u_iu_i}{2}\overline{(\text{Cont.})}_j^{i} \end{aligned} \tag{4.18}$$

Although $(\text{Cont.})_j=0$, we have proven that the divergence form of the convective term is fully conservative in the energy equation, we also note that the resulting Beam-Warming linearization applied to the convective term is not skew-symmetric, and hence is not energy-conserving.

4.1.3 Approximate Factorization

Details in the implementation of Step 1 of the fractional step method will be explained here. We employ the approximate factorization used by Kim *et al.* [7] to avoid the inversion of the large and sparse matrix A defined in Eq. (4.4). The use of the "delta form" ensures a second-order temporal accuracy in the resulting approximate factorization as we will see. We first subtract $A\mathbf{u}^n$ on both sides of Eq. (4.3):

$$A\delta\mathbf{u}^* = -A\mathbf{u}^n + \boldsymbol{r} + \boldsymbol{mbc} \equiv \boldsymbol{R}, \tag{4.19}$$

where:

$$\delta\mathbf{u}^* = \mathbf{u}^* - \mathbf{u}^n. \tag{4.20}$$

We recall that $A = \frac{1}{\Delta t}M + N - \frac{\nu}{2}L$. We replace $\mathbf{M} \equiv N - \frac{\nu}{2}L$ such that Eq. (4.19) in the 3-D case is:

$$\begin{bmatrix} \delta\overline{V}^1 + \Delta t\mathbf{M}_{11} & \Delta t\mathbf{M}_{12} & \Delta t\mathbf{M}_{13} \\ \Delta t\mathbf{M}_{21} & \delta\overline{V}^2 + \Delta t\mathbf{M}_{22} & \Delta t\mathbf{M}_{23} \\ \Delta t\mathbf{M}_{31} & \Delta t\mathbf{M}_{32} & \delta\overline{V}^3 + \Delta t\mathbf{M}_{33} \end{bmatrix} \begin{bmatrix} \delta u_1^* \\ \delta u_2^* \\ \delta u_3^* \end{bmatrix} = \begin{bmatrix} \Delta t R_1 \\ \Delta t R_2 \\ \Delta t R_3 \end{bmatrix} \tag{4.21}$$

We perform an approximate factorization of Eq. (4.21), which preserves second-order temporal accuracy. This LU approximation together with the "delta form" proposed by Kim *et al.* results in $\mathcal{O}(\Delta t^2)$ errors.

$$\begin{bmatrix} \delta\overline{V}^1 + \Delta t\mathbf{M}_{11} & 0 & 0 \\ \Delta t\mathbf{M}_{21} & \delta\overline{V}^2 + \Delta t\mathbf{M}_{22} & 0 \\ \Delta t\mathbf{M}_{31} & \Delta t\mathbf{M}_{32} & \delta\overline{V}^3 + \Delta t\mathbf{M}_{33} \end{bmatrix} \begin{bmatrix} I & \Delta t\mathbf{M}_{12}/\delta\overline{V}^1 & \Delta t\mathbf{M}_{13}/\delta\overline{V}^1 \\ 0 & I & \Delta t\mathbf{M}_{23}/\delta\overline{V}^2 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \delta u_1^* \\ \delta u_2^* \\ \delta u_3^* \end{bmatrix} = \begin{bmatrix} \Delta t R_1 \\ \Delta t R_2 \\ \Delta t R_3 \end{bmatrix}$$

One can simply multiply the factorized LU approximation matrices and compare the result with the original matrix in Eq. (4.21) to arrive at the following second-order temporal error:

$$\mathcal{O}(\Delta t^2) = \begin{pmatrix} \Delta t^2\mathbf{M}_{11}\mathbf{M}_{12}\delta u_2^*/\delta\overline{V}^1 + \Delta t^2\mathbf{M}_{11}\mathbf{M}_{13}\delta u_3^*/\delta\overline{V}^1 \\ \Delta t^2\mathbf{M}_{21}\mathbf{M}_{12}\delta u_2^*/\delta\overline{V}^1 + \Delta t^2\mathbf{M}_{21}\mathbf{M}_{13}\delta u_3^*/\delta\overline{V}^1 + \Delta t^2\mathbf{M}_{22}\mathbf{M}_{23}\delta u_3^*/\delta\overline{V}^2 \\ \Delta t^2\mathbf{M}_{31}\mathbf{M}_{12}\delta u_2^*/\delta\overline{V}^1 + \Delta t^2\mathbf{M}_{31}\mathbf{M}_{13}\delta u_3^*/\delta\overline{V}^1 + \Delta t^2\mathbf{M}_{32}\mathbf{M}_{23}\delta u_3^*/\delta\overline{V}^2 \end{pmatrix} \tag{4.22}$$

The following are the steps used in solving for $\mathbf{u}^*$ in Eq. (4.3). Note that the intermediate term $\delta\mathbf{u}^{**}$ needs

to be introduced due to the nature of the LU approximation.

$$\left(\delta\overline{V}^1 + \Delta t\mathbf{M}_{11}\right)\delta u_1^{**} = \Delta t R_1 \tag{4.23a}$$

$$\left(\delta\overline{V}^2 + \Delta t\mathbf{M}_{22}\right)\delta u_2^{**} = \Delta t R_2 - \Delta t\mathbf{M}_{21}\delta u_1^{**} \tag{4.23b}$$

$$\left(\delta\overline{V}^3 + \Delta t\mathbf{M}_{33}\right)\delta u_3^{*} = \Delta t R_3 - \Delta t\mathbf{M}_{31}\delta u_1^{**} - \Delta t\mathbf{M}_{32}\delta u_2^{**} \tag{4.23c}$$

$$\delta u_2^{*} = \delta u_2^{**} - \left(\Delta t\mathbf{M}_{23}/\delta\overline{V}^2\right)\delta u_3^{*} \tag{4.23d}$$

$$\delta u_1^{*} = \delta u_1^{**} - \left(\Delta t\mathbf{M}_{12}/\delta\overline{V}^1\right)\delta u_2^{*} - \left(\Delta t\mathbf{M}_{13}/\delta\overline{V}^1\right)\delta u_3^{*} \tag{4.23e}$$

$$u_i^{*} = u_i^{n} + \delta u_i^{*}\ , \qquad (i = 1, 2, 3) \tag{4.23f}$$

Kim *et al.*'s [7] approximate factorization for solving the intermediate velocity $\mathbf{u}^*$ results in the inversion of only the diagonal submatrices instead of the original matrix A, which also contains off-diagonal terms and is considerably more difficult to invert. This is a significant reduction in computational cost and memory since A is usually large and sparse and the inversion has to be done once at every time-step.

4.2 Stokes Equations

For a first test, we take a case where the convective terms are identically zero and the velocity vector is initialized with the following components:

$$\mathbf{u} = \begin{pmatrix} \sin\left(\dfrac{2\pi x}{L_x}\right)\cos\left(\dfrac{2\pi y}{L_y}\right)\sin\left(\dfrac{2\pi z}{L_z}\right) \\ \sin\left(\dfrac{4\pi x}{L_x}\right)\sin\left(\dfrac{2\pi y}{L_y}\right)\sin\left(\dfrac{2\pi z}{L_z}\right) \\ \cos\left(\dfrac{4\pi x}{L_x}\right)\cos\left(\dfrac{4\pi y}{L_y}\right)\cos\left(\dfrac{2\pi z}{L_z}\right) \end{pmatrix}. \tag{4.24}$$

4.2.1 Temporal Convergence

Temporal convergence is carried out on a 32^3 uniform grid with varying computational time steps. The reference "solution" is obtained by using a sufficiently fine time step of $\Delta t = 10^{-5}$. Pressure terms are evaluated at $n+1/2$ to reproduce the order of temporal accuracy presented in Kim's paper [7]. Measurements

are made at $T = 0.1$.

Table 4.1: Stokes equations: Temporal errors for the u velocity.

Δt	L_1	L_2	L_∞
0.1E−2	1.06173E-06	1.45226E-06	4.11000E-06
0.5E−3	2.65836E-07	3.63235E-07	1.02000E-06
0.25E−3	6.62966E-08	9.04434E-08	2.70000E-07
Order	2.00	2.00	1.96

Table 4.2: Stokes equations: Temporal errors for the v velocity.

Δt	L_1	L_2	L_∞
0.1E−2	3.66445E-07	5.03251E-07	1.44600E-06
0.5E−3	9.15660E-08	1.25590E-07	3.66000E-07
0.25E−3	2.42005E-08	3.28765E-08	1.03000E-07
Order	1.96	1.96	1.90

Table 4.3: Stokes equations: Temporal errors for the w velocity.

Δt	L_1	L_2	L_∞
0.1E−2	6.94010E-07	9.50947E-07	2.66900E-06
0.5E−3	1.73817E-07	2.38159E-07	6.67000E-07
0.25E−3	4.30412E-08	5.89074E-08	1.66000E-07
Order	2.00	2.00	2.00

Table 4.4: Stokes equations: Temporal errors for the pressure.

Δt	L_1	L_2	L_∞
0.1E−2	1.25663E-06	1.71432E-06	4.80727E-06
0.5E−3	3.08830E-07	4.21676E-07	1.19330E-06
0.25E−3	7.68541E-08	1.04561E-07	3.14380E-07
Order	2.01	2.01	1.96

4.2.2 Spatial Convergence

Method of Manufactured Solutions (MMS)

It is not always easy to isolate the purely spatial or temporal error for a numerical scheme. However, we will attempt to isolate the spatial errors by use of the Method of Manufactured Solutions. Neglecting higher order terms, discretization error for any scheme can be written as:

$$\|\varepsilon_{h_x}^{h_t}\| = c_x h_x^p + c_t h_t^q, \tag{4.25}$$

where p and q are the orders of convergence in space and time, respectively, while c_x and c_t are constant coefficients that do not depend on the grid spacing or time step. Presenting the case for spatial refinement from mesh configuration of 16 to 64 with a refinement factor of $1/2$ in space and keeping $\Delta t =$ constant, the errors are as follows:

$$\|\varepsilon_{16}^{h_t}\| = c_x h_{16}^p + c_t h_t^q, \tag{4.26}$$

$$\|\varepsilon_{32}^{h_t}\| = c_x h_{32}^p + c_t h_t^q, \tag{4.27}$$

$$\|\varepsilon_{64}^{h_t}\| = c_x h_{64}^p + c_t h_t^q. \tag{4.28}$$

We notice that $c_t h_t^q$ is a constant term that can be eliminated by doing some arithmetic between the equations. We finally get the following formula for determining the spatial order:

$$p = \frac{\ln\left(\dfrac{\|\varepsilon_{r^2 h_x}^{h_t}\| - \|\varepsilon_{r h_x}^{h_t}\|}{\|\varepsilon_{r h_x}^{h_t}\| - \|\varepsilon_{h_x}^{h_t}\|}\right)}{\ln(r)}, \tag{4.29}$$

where r is the constant refinement factor and (h_t, h_x) are the time step and grid spacing at the coarsest level.

The following convergence tests makes use of Eq. (4.29) with a refinement factor $r = 1/2$ and a constant time step of $\Delta t = 0.1\text{E}{-2}$. Measurements are made at $T = 0.1$ as in [4, 14, 7].

Table 4.5: Stokes equations: Spatial errors for the u velocity.

Mesh	L_1	L_2	L_∞
16^3	3.88652E-03	5.33826E-03	1.94406E-02
32^3	9.54173E-04	1.31298E-03	4.98601E-03
64^3	2.37275E-04	3.26906E-04	1.26161E-03
Order	2.01	2.01	1.97

Table 4.6: Stokes equations: Spatial errors for the v velocity.

Mesh	L_1	L_2	L_∞
16^3	7.66764E-03	1.05784E-02	2.77226E-02
32^3	1.92405E-03	2.61358E-03	7.32688E-03
64^3	4.79762E-04	6.51415E-04	1.85406E-03
Order	1.99	2.01	1.95

Table 4.7: Stokes equations: Spatial errors for the w velocity.

Mesh	L_1	L_2	L_∞
16^3	1.01922E-02	1.36347E-02	3.69281E-02
32^3	2.47951E-03	3.35161E-03	9.78369E-03
64^3	6.18401E-04	8.34346E-04	2.46126E-03
Order	2.02	2.01	1.95

Table 4.8: Stokes equations: Spatial errors for the pressure.

Mesh	L_1	L_2	L_∞
16^3	6.75709E-02	8.53005E-02	2.46821E-01
32^3	1.64249E-02	2.10507E-02	6.66404E-02
64^3	4.09184E-03	5.24921E-03	1.68809E-02
Order	2.02	2.01	1.93

4.3 Navier-Stokes Equations

We will now proceed to test the full Navier-Stokes equations. The velocity field is initialized to the following divergence-free field:

$$\mathbf{u} = \begin{pmatrix} \cos\left(\frac{2\pi x}{L}\right)\sin\left(\frac{2\pi y}{L}\right)\sin\left(\frac{2\pi z}{L}\right) \\ \sin\left(\frac{2\pi x}{L}\right)\cos\left(\frac{2\pi y}{L}\right)\sin\left(\frac{2\pi z}{L}\right) \\ -2\sin\left(\frac{2\pi x}{L}\right)\sin\left(\frac{2\pi y}{L}\right)\cos\left(\frac{2\pi z}{L}\right) \end{pmatrix} \tag{4.30}$$

where L is the size of the domain in x, y and z.

4.3.1 Temporal Convergence

Temporal convergence is carried out on 32^3 uniform grid with varying computational time steps. The reference "solution" is obtained by again by choosing a sufficiently fine time step of $\Delta t = 10^{-5}$. Measurements are made at $T = 0.1$. The following results are obtained using the Beam-Warming linearization for the convective terms.

Table 4.9: Navier-Stokes equations: Temporal errors for the u velocity.

Δt	L_1	L_2	L_∞
0.1E−2	6.70941E-07	9.17701E-07	2.59000E-06
0.5E−3	1.68593E-07	2.29564E-07	6.50000E-07
0.25E−3	4.23217E-08	5.73956E-08	1.70000E-07
Order	1.99	1.99	1.96

Table 4.10: Navier-Stokes equations: Temporal errors for the v velocity.

Δt	L_1	L_2	L_∞
0.1E−2	2.68110E-06	3.67270E-06	1.03000E-05
0.5E−3	6.71190E-07	9.18492E-07	2.58000E-06
0.25E−3	1.67915E-07	2.29559E-07	6.50000E-07
Order	1.99	1.99	1.99

Table 4.11: Navier-Stokes equations: Temporal errors for the w velocity.

Δt	L_1	L_2	L_∞
0.1E−2	3.35443E-06	4.59075E-06	1.29300E-05
0.5E−3	8.43181E-07	1.14947E-06	3.28000E-06
0.25E−3	2.10828E-07	2.87176E-07	8.20000E-07
Order	1.99	1.99	1.98

Table 4.12: Navier-Stokes equations: Temporal errors for the pressure.

Δt	L_1	L_2	L_∞
0.1E−2	3.55313E-06	4.84519E-06	1.41653E-05
0.5E−3	8.76065E-07	1.19415E-06	3.62238E-06
0.25E−3	2.21976E-07	2.99268E-07	9.64810E-07
Order	2.00	2.00	1.93

4.4 Grid-Stretching: Navier-Stokes Equations

We will now proceed to test the Navier-Stokes equations on a non-uniform mesh to further validate the accuracy results from the previous section. Again, we use a divergence-free velocity field:

$$\mathbf{u} = \begin{pmatrix} \sin\left(\frac{2\pi x}{L}\right)\cos\left(\frac{2\pi y}{L}\right)\cos\left(\frac{2\pi z}{L}\right) \\ \cos\left(\frac{2\pi x}{L}\right)\sin\left(\frac{2\pi y}{L}\right)\cos\left(\frac{2\pi z}{L_z}\right) \\ -2\cos\left(\frac{2\pi x}{L}\right)\cos\left(\frac{2\pi y}{L}\right)\sin\left(\frac{2\pi z}{L}\right) \end{pmatrix} \tag{4.31}$$

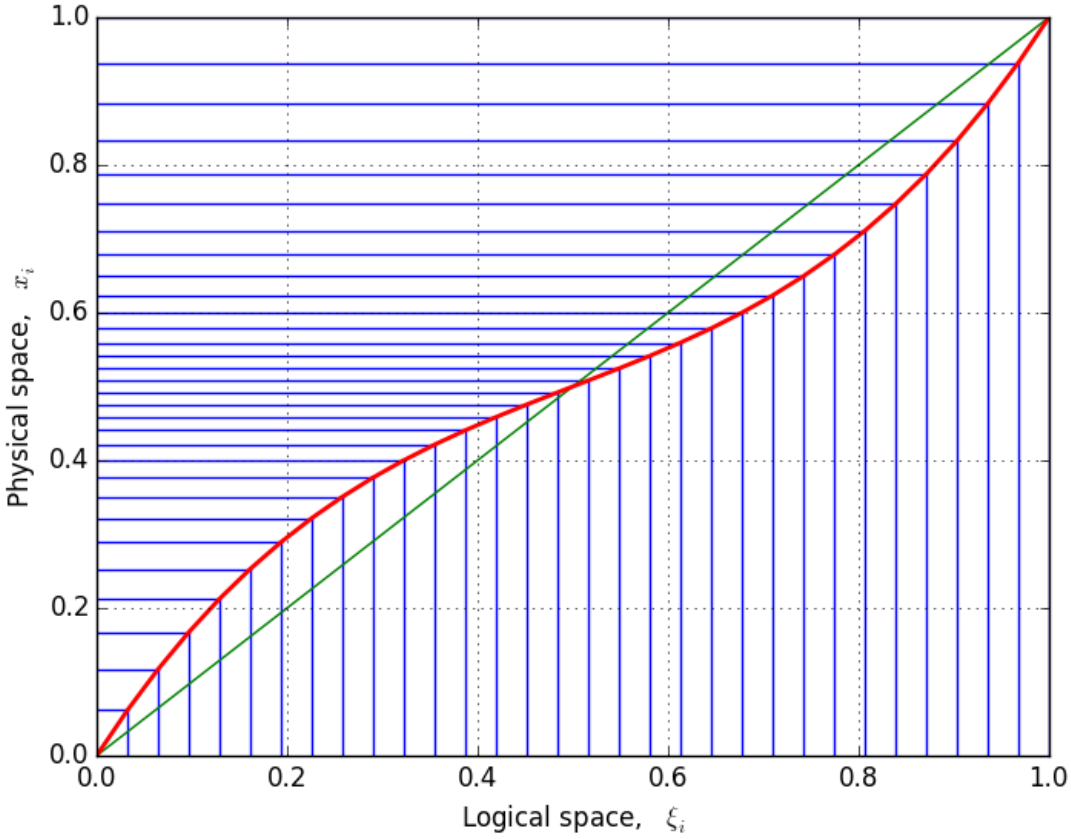


Figure 4.2: Mesh stretching in each dimension

The equation used for stretching the mesh is:

$$x_i = \xi_i + A(x_c - L\xi_i)(1 - \xi_i)\xi_i \tag{4.32}$$

where the chosen coefficients are $L = 2$, $A = 1$ and $x_c = 1$. The coefficient A is generally referred to as the *strength*, because it concentrates on points inside the domain (if $A > 0$) or towards the boundary (if $A < 0$). The term $(1 - \xi_i)\xi_i$ ensures that the end points of the grid remain at $\{0, 1\}$. Finally, $x_c - L\xi_i$ makes the formula switch sign at the points x_c/L.

In all tests below, we scaled all three axes with Eq. (4.32) in an attempt to find bugs in all three directions. We actually did it incrementally to be more rigorous in debugging (by scaling in one direction, then two, then three), even though the we did not reveal the numerical results for these tests here, since they are simplified examples of the more general case.

4.4.1 Temporal Convergence

Temporal convergence is carried out on 32^3 non-uniform grid with varying computational time steps. The reference "solution" is obtained by using a sufficiently fine time step of $\Delta t = 10^{-5}$. Measurements are made at $T = 0.1$.

Table 4.13: Non-uniform mesh: Temporal errors for the u velocity.

Δt	L_1	L_2	L_∞
0.250E−3	4.38690E-08	5.97704E-08	2.00000E-07
0.200E−3	3.60916E-08	4.81427E-08	1.60000E-07
0.125E−3	1.18316E-08	1.65942E-08	7.00000E-08
Order	1.96	1.91	1.55

Table 4.14: Non-uniform mesh: Temporal errors for the v velocity.

Δt	L_1	L_2	L_∞
0.250E−3	1.76539E-07	2.39717E-07	6.70000E-07
0.200E−3	1.15187E-07	1.56096E-07	4.70000E-07
0.125E−3	4.38866E-08	5.98077E-08	1.90000E-07
Order	2.01	2.00	1.83

Table 4.15: Non-uniform mesh: Temporal errors for the w velocity.

Δt	L_1	L_2	L_∞
0.250E−3	2.19949E-07	2.98545E-07	8.70000E-07
0.200E−3	1.47276E-07	1.98929E-07	6.30000E-07
0.125E−3	5.50014E-08	7.47811E-08	2.40000E-07
Order	2.01	2.01	1.88

Table 4.16: Non-uniform mesh: Temporal errors for the pressure.

Δt	L_1	L_2	L_∞
0.250E−3	2.68324E-07	3.43407E-07	9.93270E-07
0.200E−3	1.73002E-07	2.21560E-07	8.73150E-07
0.125E−3	6.95216E-08	8.68382E-08	2.77740E-07
Order	1.94	1.98	1.93

4.4.2 Spatial Convergence: Convective Operator

We can see that the (u, v) components of the velocity have very similar orders and w is slightly worse. This is because of the periodicity in the initial condition for the (u, v) components, which are virtually the same,

Table 4.17: Non-uniform mesh: Spatial errors in the convective term for u.

Mesh	L_1	L_2	L_∞
32^3	4.66191E-02	6.85506E-02	3.03591E-01
64^3	1.22076E-02	1.76417E-02	8.81622E-02
Order	1.93	1.95	1.78

Table 4.18: Non-uniform mesh: Spatial errors in convective term for v velocity.

Mesh	L_1	L_2	L_∞
32^3	4.66191E-02	6.85506E-02	3.03591E-01
64^3	1.22076E-02	1.76417E-02	8.81622E-02
Order	1.93	1.95	1.78

Table 4.19: Non-uniform mesh: Spatial errors in convective term for w velocity.

Mesh	L_1	L_2	L_∞
32^3	1.24354E-01	1.92789E-01	9.94647E-01
64^3	3.28181E-02	5.01937E-02	2.56643E-01
Order	1.92	1.94	1.95

while the w component has an extra factor of 2, which basically increases its period by a factor of two, thus keeping it out of sync with the other two.

4.4.3 Spatial Convergence: Viscous Operator

Table 4.20: Non-uniform mesh: Spatial errors in the viscous term for u.

Mesh	L_1	L_2	L_∞
32^3	3.12108E-01	4.48803E-01	1.99704E+00
64^3	8.04149E-02	1.17428E-01	6.38208E-01
Order	1.95	1.93	1.64

Table 4.21: Non-uniform mesh: Spatial errors in the viscous term for v.

Mesh	L_1	L_2	L_∞
32^3	6.63976E-01	9.92639E-01	3.89627E+00
64^3	1.71557E-01	2.53897E-01	1.03040E+00
Order	1.95	1.96	1.91

Note that different MMS functions were used for testing the diffusive operator. Each of the velocity components has a different periodicity in one of the axes such that convergence results vary, unlike in the convective term. In the tests, we noticed that L_∞ suffers when functions are more oscillatory (w velocity is initialized with a

Table 4.22: Non-uniform mesh: Spatial errors in the viscous term for w.

Mesh	L_1	L_2	L_∞
32^3	1.04915E+00	1.66823E+00	8.70721E+00
64^3	2.66713E-01	4.51114E-01	3.54487E+00
Order	1.97	1.88	1.29

function of periodicity 4π).

We are interested to check if there were bugs in the divergence and gradient operators (both for the staggered and collocated grid) because numerous interpolations between face areas are required in the construction of those operators and if the mesh size were all uniform, it is not possible to detect a large class of issues. Again, as a reminder, the viscous term results from the convolution of the discrete divergence and gradient operator on the staggered grid, while the Laplacian in the Poisson equation results from the convolution of the discrete divergence and gradient operator on the collocated grid. Hence such spatial checks as above would have required proper formulation in all the divergence and gradient operators. As it seems, the spatial and temporal convergence results are convincing and give reasons to believe that the implementation is correct.

Part II

Immersed Boundary Projection Method

Chapter 5

Immersed Boundary Method: Projection Approach

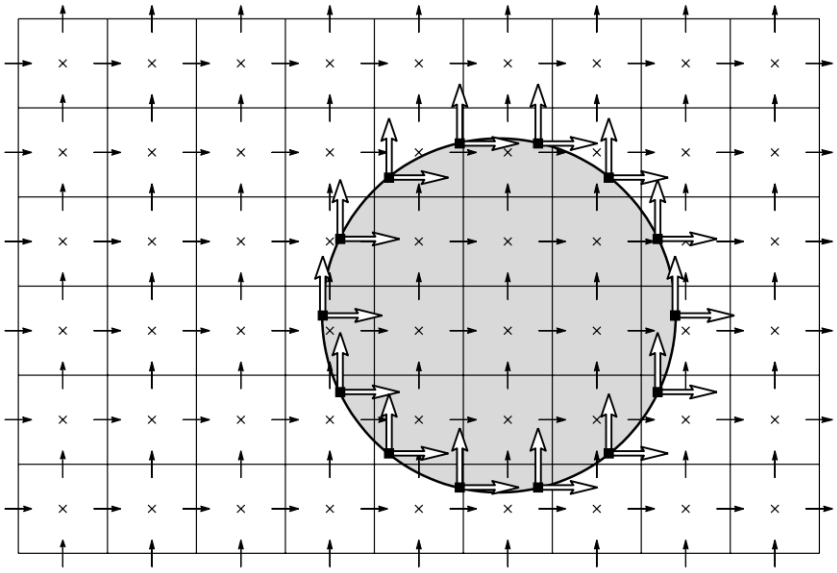


Figure 5.1: Distribution of Lagrangian points on Eulerian grid (Taira and Colonius [14]).

We first review how Taira and Colonius [14] incorporated Perot's LU decomposition in their formulation of the Immersed Boundary Projection Method (IBPM) and we will talk about how it can be enhanced using Brown *et al.*'s [15] higher order pressure update schemes. A derivation of Taira and Colonius's IBPM is given in this section.

We have the underlying spatial grid or the computational domain $\mathcal{D}$ as well as the Lagrangian points, ξ_k that lie on the immersed object's surface $\partial\mathcal{B}$. As one can imagine, it is important for both parties to communicate and exchange information from $\partial\mathcal{B}$ to $\mathcal{D}$ and vice versa. This is done by adding extra terms to the Navier-Stokes equations that represent these transfers. The new system of equations is:

$$\begin{cases} \dfrac{\partial u}{\partial t} + u \cdot \nabla u = -\nabla p + \dfrac{1}{Re}\nabla^2 u + \displaystyle\int_s f(\xi(s,t))\delta(\xi - x)\mathrm{d}s, \\ \nabla \cdot u = 0, \\ u(\xi(s,t)) = \displaystyle\int_s u(\xi(s,t))\delta(\xi - x)\mathrm{d}s = u_B(\xi(s,t)). \end{cases} \tag{5.1}$$

Some further remarks on the above equations:

- The immersed object's boundary ∂B is parameterized by s and moves with a velocity of $u_B(\xi(s,t))$.
- f is the boundary force (treated like a body force in the Navier-Stokes equations).
- Taira and Colonius [14] also mentioned that the distance between the Lagrangian points should be $\approx$ the grid size of the computational domain, for stability reasons. Also no two Lagrangian points should coincide (it leads to singularities in some of the operators we will define).

In Taira and Colonius's implementation, they solved in terms of velocity fluxes, q instead of velocity u. If we discretize (5.1) completely in space and time and replace velocity with velocity fluxes, the resulting system can be written in block matrix form as follows:

$$\begin{bmatrix} A & G & -H \\ D & 0 & 0 \\ E & 0 & 0 \end{bmatrix} \begin{pmatrix} q^{n+1} \\ \phi \\ f \end{pmatrix} = \begin{pmatrix} r^n \\ 0 \\ u_B^{n+1} \end{pmatrix} + \begin{pmatrix} bc_1 \\ bc_2 \\ 0 \end{pmatrix} \tag{5.2}$$

We make use of certain properties of the submatrices ($D = -G^T$ and $Hf = -E^T\tilde{f}$), which allows us to obtain the following system of equations:

$$\begin{bmatrix} A & G & E^T \\ G^T & 0 & 0 \\ E & 0 & 0 \end{bmatrix} \begin{pmatrix} q^{n+1} \\ \phi \\ \tilde{f} \end{pmatrix} = \begin{pmatrix} r^n \\ 0 \\ u_B^{n+1} \end{pmatrix} + \begin{pmatrix} bc_1 \\ bc_2 \\ 0 \end{pmatrix}, \tag{5.3}$$

which can take the form of the KKT conditions encountered in quadratic minimization problems (seen later). Taira and Colonius follow closely the fractional step based on Perot's LU approximation in solving the system above. If we let:

$$Q \equiv \begin{bmatrix} G & E^T \end{bmatrix}, \quad \lambda \equiv \begin{pmatrix} \phi \\ \tilde{f} \end{pmatrix}, \quad r_2 \equiv \begin{pmatrix} -bc_2 \\ u_B^{n+1} \end{pmatrix},$$

we can rewrite (5.3) as:

$$\begin{bmatrix} A & Q \\ Q^T & 0 \end{bmatrix} \begin{pmatrix} q^{n+1} \\ \lambda \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}. \tag{5.4}$$

After performing a block LU decomposition on (5.4), we get:

$$\begin{bmatrix} A & 0 \\ Q^T & -Q^T B^N Q \end{bmatrix} \begin{bmatrix} I & B^N Q \\ 0 & I \end{bmatrix} \begin{pmatrix} q^{n+1} \\ \lambda \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} + \begin{pmatrix} -\dfrac{\Delta t^N}{2^N}(LM^{-1})^N Q\lambda \\ 0 \end{pmatrix}. \tag{5.5}$$

However, the system is usually not solved using (5.5), since it would require inverting very large matrices.

We can also write the system in a more manageable manner as follows:

$$Aq^* = r_1, \qquad \text{(Get intermediate velocity flux, } q^*\text{)} \tag{5.6}$$

$$Q^T B^N Q\lambda = Q^T q^* - r_2, \qquad \text{(Solve Poisson equation for } \lambda\text{)} \tag{5.7}$$

$$q^{n+1} = u^* - B^N Q\lambda. \qquad \text{(Projection step)} \tag{5.8}$$

5.1 Dimensional Consistency

In checking for dimensional consistency in the equations, much can be learned from Taira and Colonius's implementation. There exist two types of equations in their 2-D implementation: (1) the original equations and (2) the scaled equations. Questions such as what are the dimensions of ϕ, why did they solve the system in terms of velocity fluxes instead of velocity itself will be answered.

Scaling Matrices

There are two kinds of operators that exist in [14]: (1) with "hats" (e.g. $\hat{A}$) and (2) without "hats" (e.g. A). Those with "hats" represent the original operators with the usual dimensions, while those without have already been scaled by matrices such as R and $\hat{M}$:

$$R \equiv \begin{pmatrix} \Delta y_j & 0 \\ 0 & \Delta x_i \end{pmatrix} \qquad \hat{M} \equiv \begin{pmatrix} \frac{1}{2}(\Delta x_{i-1} + \Delta x_i) & 0 \\ 0 & \frac{1}{2}(\Delta y_{j-1} + \Delta y_j) \end{pmatrix} \tag{5.9}$$

Both are scaling matrices with dimensions of length. For simplicity, SI units are used to represent the dimensions. All dimensions that are presented below have already been divided by the dimensions of density (which is kg/m^3).

5.1.1 Original Equations

$$\frac{u^{n+1}-u^n}{\Delta t}+\frac{3}{2}\hat{N}(u^n)-\frac{1}{2}\hat{N}(u^{n-1})=-\hat{G}\phi+\frac{1}{2}(\hat{L}(u^{n+1})+\hat{L}(u^n))+\hat{H}(f)+b\hat{c_1}$$

$$\hat{A}(u^{n+1})+\hat{G}\phi-\hat{H}(f)=\underbrace{\left[\frac{1}{\Delta t}I+\frac{1}{2}\hat{L}\right]u^n-\frac{3}{2}\hat{N}(u^n)+\frac{1}{2}\hat{N}(u^{n-1})}_{\hat{r}^n}+b\hat{c_1} \tag{5.10}$$

$$\hat{D}(u^{n+1})=bc_2 \tag{5.11}$$

$$\hat{E}(u^{n+1})=u_B^{n+1} \tag{5.12}$$

This is the fully discretized form of the original equations that was presented in Taira's paper. Lets analyze each term:

1: $\frac{u^{n+1}-u^n}{\Delta t}$: the dimensions of this first term are simply $\left[\frac{u^{n+1}-u^n}{\Delta t}\right]_{units} \equiv m/s^2$. This also means that each and every term in Eq. (5.10) should be m/s^2.

2: $\hat{A}$: Mentioned in Eq. (47) in [14], $\hat{A} \equiv \frac{1}{\Delta t}I-\frac{1}{2}\hat{L}$. Therefore the dimensions of the operator have to be:

$$[\hat{A}]_{units} \equiv 1/s. \tag{5.13}$$

We can also back-check to see that $[\hat{A}(u)]_{units} \equiv m/s^2$, which is again consistent with previous results.

3: $\hat{G}$: We first need to find out the dimensions of ϕ. Does it have the same dimensions as pressure? Direct quote from Taira's paper, " ... discrete pressure is denoted by ϕ without any superscript for its time level, as we regard pressure as a Lagrange multiplier. ", "... we use ϕ as a simple representation of the pressure variable ...". This suggests that ϕ does indeed hold the same dimensions as pressure and pressure's dimensions (divided by density) is m^2/s^2. This hypothesis will be not be left unchecked, as we will eventually prove in later sections that this dimension is necessary for consistency. Given that $[\hat{G}\phi]_{units}$ has to be m/s^2, the dimensions of the operator has to be:

$$[\hat{G}]_{units} \equiv 1/m. \tag{5.14}$$

4: $\hat{L}$: This is simple as we see from once again Eq. (47) in [14], $\hat{A} \equiv \frac{1}{\Delta t}I-\frac{1}{2}\hat{L}$. Therefore $\hat{L}$ and $\hat{A}$

must have same dimensions:

$$[\hat{L}]_{units} \equiv 1/s. \tag{5.15}$$

We can more rigorously check this result by realizing that $\hat{L}$'s dimensions consists of one partial derivative $[\partial/\partial x_i]_{units} \equiv 1/m$, one divergence $[\partial/\partial x_j]_{units} \equiv 1/m$ and a dynamic viscosity $[\mu]_{units} \equiv Pa.s \equiv m^2/s$, such that

$$\begin{aligned}[\hat{L}]_{units} &\equiv (1/m)(1/m)(m^2/s) \\ &= 1/s,\end{aligned} \tag{5.16}$$

with dimensions of $[\hat{L}(u)]_{units} \equiv m/s^2$ being consistent.

5: $\hat{H}$: In the Navier-Stokes equations, f has the units of acceleration or force per unit mass m/s^2. For consistency, $\hat{H}(f)$ must have dimensions of m/s^2, giving us:

$$[\hat{H}]_{units} \equiv dimensionless. \tag{5.17}$$

6: $\hat{D}$: A reasonable assumption is

$$[\hat{D}]_{units} \equiv m. \tag{5.18}$$

and this will eventually be proven to be correct in later sections. The above tells us that $[bc_2]_{units} \equiv m^2/s$. This will be important to us later on.

7: $\hat{E}$: The following has to be true for Eq. (5.12) to be consistent:

$$[\hat{E}]_{units} \equiv dimensionless. \tag{5.19}$$

All the above material will be ascertained when we look at the dimensions of the scaled equations.

5.1.2 Scaled Equations

$$q = R(u) \tag{5.20}$$

$$\frac{q^{n+1}-q^n}{\Delta t} + \frac{3}{2}N(q^n) - \frac{1}{2}N(q^{n-1}) = -G\phi + \frac{1}{2}(L(q^{n+1}) + L(q^n)) + H(f) + bc_1$$

$$A(q^{n+1}) + G\phi - H(f) = \underbrace{\left[\frac{1}{\Delta t}M + \frac{1}{2}L\right]q^n - \frac{3}{2}N(q^n) + \frac{1}{2}N(q^{n-1})}_{r^n} + bc_1 \tag{5.21}$$

$$D(q^{n+1}) = bc_2 \tag{5.22}$$

$$E(q^{n+1}) = u_B^{n+1} \tag{5.23}$$

1: q : In Taira's paper, the following equation is mentioned without index, $q = Ru$. As such, q is to be understood as a velocity flux with dimensions

$$\begin{aligned}[q]_{units} &\equiv (m)(m/s) \\ &= m^2/s. \end{aligned} \tag{5.24}$$

2: $\frac{q^{n+1}-q^n}{\Delta t}$:

$$\left[\frac{q^{n+1}-q^n}{\Delta t}\right]_{units} \equiv m^2/s^2. \tag{5.25}$$

This means that, for consistency, every term in Eq. (5.21) should have the dimensions of m^2/s^2.

3: A: In Eq. (50) from Taira's paper [14], we were given the scaling of A where $A = \hat{M}\hat{A}R^{-1}$

$$\begin{aligned}[A]_{units} &\equiv (m)(1/s)(1/m) \\ &= 1/s. \end{aligned} \tag{5.26}$$

We can also back-check to see that $[A(q)]_{units} \equiv m^2/s^2$ which is consistent.

4: G: Also in Eq. (50) [14], we see the scaling of G where $G = \hat{M}\hat{G}$ giving us

$$\begin{aligned}[G]_{units} &\equiv (m)(1/m) \\ &= dimensionless. \end{aligned} \tag{5.27}$$

This is further evidenced by quoting directly from [14] "Both operators (referring to gradient and divergence) can be scaled appropriately so that the entries consist solely of ± 1 by introducing R and

$\hat{M}$." As it seems, this validated the dimensions of ϕ when we hypothesized that $[\phi]_{units} = [p]_{units}$. This is because if G is dimensionless, dimensional consistency in Eq. (5.21) requires $[\phi]_{units} \equiv m^2/s^2$.

5: L: The following is given without indexing in Taira's paper $A = \frac{1}{\Delta t}M - \frac{1}{2}L$ where $M \equiv \hat{M}R^{-1}$. Both equations appear within some text slightly after Eq. (50) [14]. Hence, $[M]_{units} \equiv dimensionless$ and once again we see that

$$\begin{aligned} [L]_{units} &\equiv [A]_{units} \\ &= 1/s. \end{aligned} \tag{5.28}$$

We also see that $[L(q)]_{units} \equiv m^2/s^2$ which is consistent.

6: H: From Eq. (51) in [14], we see the scaling of H, where $H \equiv \hat{M}\hat{H}$, giving us:

$$[H]_{units} \equiv m \tag{5.29}$$

As a result, $H(f)$ has dimensions of m^2/s^2 which is consistent.

7: D: We come to the scaled continuity equations and noticed that throughout Taira's paper, bc_2 has no "hats" whether in the original or scaled equations. This must mean that bc_2 did not change in dimensions. This can be further confirmed with the following scaling provided in Eq. (50) [14] that $D \equiv \hat{D}R^{-1}$, which gives us $[D]_{units} \equiv (m)(1/m) = dimensionless$. This is further confirmed by the statement made by Taira that his G and D are ± 1 together with his Eq. (50) where $D \equiv \hat{D}R^{-1} = -G^T$. This is proof that his $[\hat{D}]_{units}$ is indeed $= m$. Hence

$$[D]_{units} \equiv dimensionless. \tag{5.30}$$

Once again, we back check to see that the units of $D(q)$ is m^2/s which means that bc_2's dimensions truly didn't change from original to scaled equations.

8: E: The checking of dimensions revealed certain keywords mentioned in Taira's paper when he said "It should be observed that E and H are symmetric up to a constant ...". Let's check the dimensions of E. It is given again in Eq. (51) [14] that $E \equiv \hat{E}R^{-1}$, implying that $[E]_{units} \equiv 1/m$. This is further substantiated when we look at Eq. (5.23), where $E(q) = u_B$ and for consistency $[E]_{units}$ has to be $1/m$.

$$[E]_{units} \equiv 1/m. \tag{5.31}$$

The following is the biggest revelation of it all - the need for $\tilde{f}$. E and H operators do not have the same dimensions. In 2-D, H has dimensions of m and therefore $\tilde{f}$ not only absorbs the difference in dimensions, it also absorbs any constants to make the following matrix symmetric positive definite

$$\begin{bmatrix} A & G & E^T \\ G^T & 0 & 0 \\ E & 0 & 0 \end{bmatrix}$$

This appears in Eq. (21) [14]. This is critical because without this property, we do not arrive at a symmetric system. Quoting directly from Taira's paper [14], $H(f) = -E^T(\tilde{f})$, where $f = -\frac{1}{\Delta x^2}\frac{\alpha}{\beta}\tilde{f}$.

5.2 Derivation of a KKT System

We would like to present the derivation to prove that the algebraic system of equations which results from the Navier-Stokes equations is indeed a Karush–Kuhn–Tucker (KKT) system that appears in constrained optimization problems. We are interested to see if the resulting system of equations do turn out to be a symmetric system when solved in terms of velocity fluxes (as implemented by Taira and Colonius). First, we define the following optimization problem:

$$\begin{aligned} &\min_{q^{n+1}} \left[\frac{1}{2} \left(q^{n+1}\right)^T A q^{n+1} - \left(q^{n+1}\right)^T \left(r^n + bc_1\right) \right] \\ &\text{subjected to} \quad Dq^{n+1} = bc_2. \end{aligned} \tag{5.32}$$

For convenience, the following notation is used:

$$\begin{aligned} (r^n + bc_1) &\equiv b, \\ q^{n+1} &\equiv x, \\ f(x) &\equiv \frac{1}{2} x^T A x - x^T b, \\ g(x) &\equiv Dx - bc_2 = 0. \end{aligned}$$

A necessary condition for a feasible point x^* to be a solution to this problem is that the negative gradient of f lie in the space spanned by the constraint normals, i.e., that:

$$-\nabla f(x^*) = J_g^T(x^*)\lambda^*,$$

for some $\lambda^* \in \mathbb{R}^m$, where $J_g^T(x^*)$ is the Jacobian matrix of $g(x^*)$. The Lagrangian associated with this constrained optimization problem is:

$$\begin{aligned}\mathcal{L}(x,\lambda) &= f(x) + \lambda g(x^*) \\ &= \frac{1}{2}x^T Ax - x^T b + \lambda\left(Dx - bc_2\right)\end{aligned}$$

where $\lambda \in \mathbb{R}^m$ is a vector of the Lagrange multipliers for each unknown in the original system. Consequently, the gradient of the Lagrangian function is:

$$\nabla\mathcal{L}(x,\lambda) = \begin{bmatrix}\nabla_x\mathcal{L}(x,\lambda) \\ \nabla_\lambda\mathcal{L}(x,\lambda)\end{bmatrix} = 0. \tag{5.33}$$

We shall derive the elements in Eq. 5.33 step by step.Let $P(x) := \langle x, Ax\rangle$ and let w, with $\|w\| = 1$, be a vector. We denote the directional derivative of P with respect to x in the direction w by:

$$\begin{aligned}P'(x)w &= \lim_{h\to 0}\frac{P(x+hw) - P(x)}{h} \\ &= \lim_{h\to 0}\langle x, Aw\rangle + \langle w, Ax\rangle + h\langle w, Aw\rangle \\ &= 2\langle x, Aw\rangle,\end{aligned}$$

since A is symmetric.

Let $K(x) := \langle x, b\rangle$, we have:

$$\begin{aligned}K'(x)w &= \lim_{h\to 0}\frac{K(x+hw) - K(x)}{h} \\ &= \langle b, w\rangle\end{aligned}$$

Finally, let $Q_1(x) := \langle\lambda, Dx - bc_2\rangle$. We have that:

$$\begin{aligned}Q_1'(x,\lambda)w &= \lim_{h\to 0}\frac{Q(x+hw,\lambda) - Q(x,\lambda)}{h} \\ &= \left\langle D^T\lambda, w\right\rangle\end{aligned}$$

and let $Q_2(\lambda)$, which gives:

$$\begin{aligned}Q_2'(\lambda)w &= \lim_{h\to 0}\frac{Q(x,\lambda+hw) - Q(x,\lambda)}{h} \\ &= \langle Dx - bc_2, w\rangle\end{aligned}$$

We can then use the Riesz Representation Theorem to find the gradients for the above directional derivatives,

which we can then replace into Eq. (5.33) to get the following results:

$$\nabla \mathcal{L}(x,\lambda) = \begin{bmatrix} \nabla_x \mathcal{L}(x,\lambda) \\ \nabla_\lambda \mathcal{L}(x,\lambda) \end{bmatrix} = \begin{bmatrix} Ax - b + D^T\lambda \\ Dx - bc_2 \end{bmatrix} = 0 \tag{5.34}$$

which is equivalent to:

$$\begin{bmatrix} A & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} b \\ bc_2 \end{bmatrix} \tag{5.35}$$

We can now substitute back the original variable names and multiply the second equation by -1 to match up with Navier-Stokes equations. This leaves:

$$\begin{bmatrix} A & -D^T \\ -D & 0 \end{bmatrix} \begin{bmatrix} q^{n+1} \\ \lambda \end{bmatrix} = \begin{bmatrix} r^n + bc_1 \\ -bc_2 \end{bmatrix}$$

Recall that $G = -D^T$, by construction from [31], we have that:

$$\begin{bmatrix} A & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} q^{n+1} \\ \lambda \end{bmatrix} = \begin{bmatrix} r^n + bc_1 \\ -bc_2 \end{bmatrix} \tag{5.36}$$

which is a symmetric system. A very similar analysis can be carried out to obtain the full IBPM system in (5.3) by simply adding a second Lagrange multiplier corresponding to the immersed boundary points.

Chapter 6

Sharp Immersed Boundary Projection Method

In this section, we introduce our modifications to the immersed boundary projection method presented by Taira and Colonius [14].

Original Fully-Discretized IBPM Equations

We have omitted convective term for simplicity of discussion. In any case, we'll use Morinishi's skew-symmetric form [2] for the convective term with Picard iteration for the implementation. The discretized system is:

$$M\frac{u^{n+1}-u^n}{\Delta t}+Gp^{n+1/2}=\frac{\nu}{2}L(u^{n+1}+u^n)+HW^Tf^{n+1/2}+mbc \tag{6.1}$$

$$Du^{n+1}=cbc \tag{6.2}$$

$$WEu^{n+1}=Wu_B^{n+1} \tag{6.3}$$

where the following defines the discrete operators were used above:

$$M \simeq \text{diagonal mass matrix} \tag{6.4a}$$

$$D \simeq \text{divergence operator for first-order tensor} \tag{6.4b}$$

$$G \simeq \text{gradient operator for zeroth order tensor} \tag{6.4c}$$

$$L \simeq \text{symmetric viscous operator for first-order tensor} \tag{6.4d}$$

We propose the following modifications to the existing method by Taira and Colonius [14] :

1. Solve directly in terms of velocity, u instead of velocity fluxes, q.
2. Replace the discrete Dirac delta function with multi-linear interpolation.
3. Experimenting with incremental-Lagrange multiplier projection methods as proposed by Brown *et al.*.

6.1 Approximate LU Factorization Methods

Given the previous system:

$$M\frac{u^{n+1}-u^n}{\Delta t} = -G\phi + \frac{\nu}{2}L(u^{n+1}+u^n) + HW^T f + mbc, \tag{6.5}$$

$$Du^{n+1} = cbc, \tag{6.6}$$

$$WEu^{n+1} = Wu_B^{n+1}, \tag{6.7}$$

we can perform an LU approximation on its block matrix form, to obtain:

$$\begin{pmatrix} A & 0 & 0 \\ D & -DB^NG & DB^NHW^T \\ WE & -WEB^NG & WEB^NHW^T \end{pmatrix} \begin{pmatrix} I & B^NG & -B^NHW^T \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} u^{n+1} \\ \phi \\ f \end{pmatrix} = \begin{pmatrix} r \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} mbc \\ -cbc \\ Wu_B^{n+1} \end{pmatrix} \tag{6.8}$$

where:

$$A = \frac{1}{\Delta t}M - \frac{\nu}{2}L \tag{6.9a}$$

$$r = \left[\frac{1}{\Delta t}M + \frac{\nu}{2}L\right]u^n \tag{6.9b}$$

$$B^N \approx A^{-1} = \Delta t M^{-1} + \frac{\Delta t^2}{2}(M^{-1}L)M^{-1} + \cdots + \frac{\Delta t^N}{2^{N-1}}(M^{-1}L)^{N-1}M^{-1} \tag{6.9c}$$

6.1.1 Dimensional Consistency in the Lagrangian

The KKT-type optimization problem posed for the full IBPM system is:

$$\min_{u^{n+1}} f(u^{n+1}) \quad \equiv \quad \min_{u^{n+1}} \left[\frac{1}{2}\left(u^{n+1}\right)^T A\, u^{n+1} - \left(u^{n+1}\right)^T (r+mbc)\right] \tag{6.10}$$

$$\text{subjected to} \quad -Du^{n+1} + cbc = 0 \tag{6.11}$$

$$W(Eu^{n+1} - u_B^{n+1}) = 0 \tag{6.12}$$

The constraint equation Eq. (6.11) is multiplied by negative one so that the KKT system reverts back to our governing equations in a more intuitive manner. A necessary condition for a feasible point $(u^{n+1})^*$ to be a solution (i.e. a minimum) to this problem is that the negative gradient of f lies in the space spanned by the constraint normals, i.e., that:

$$-\nabla f((u^{n+1})^*) = J_g^T((u^{n+1})^*)\lambda^*, \tag{6.13}$$

for some $\lambda^* \in \mathbb{R}^m$, where $J_g^T((u^{n+1})^*)$ is the Jacobian matrix of $g((u^{n+1})^*)$, the set of constraints in Eqs. (6.11)

and (6.12).

The Lagrangian associated with this optimization problem is:

$$\begin{aligned}\mathcal{L}(u^{n+1}, \lambda_1, \lambda_2, \ldots, \lambda_M) &= f(u^{n+1}) + \sum_{k=1}^{M} \lambda_k^T g_k(u^{n+1}) \\ &= \frac{1}{2}\left(u^{n+1}\right)^T A\, u^{n+1} - \left(u^{n+1}\right)^T (r + mbc) \\ &+ \phi\,(-Du^{n+1} + cbc) \\ &+ f^T W\,(Eu^{n+1} - u_B^{n+1}) \end{aligned} \tag{6.14}$$

where $M = 2$ (we only have two constraints) and the Lagrange multipliers we have introduced for our system are as follows

$$\lambda_1 = \phi \tag{6.15a}$$

$$\lambda_2 = f \tag{6.15b}$$

We will now perform a dimensional analysis on all the terms in the Lagrangian:

1: $f(u^{n+1})$:

$$\begin{aligned}\left[f(u^{n+1})\right]_{units} &\equiv \left[\frac{1}{2}\left(u^{n+1}\right)^T A\, u^{n+1} - \left(u^{n+1}\right)^T (r + mbc)\right]_{units} \\ &= \left[\left(u^{n+1}\right)^T A\, u^{n+1}\right]_{units} \\ &= (m/s)(m^3/s)(m/s) \\ &= m^5/s^3 \end{aligned} \tag{6.16}$$

Recall that $[A]_{units} \equiv \left[\delta\overline{V}^i/\Delta t\right]_{units} = m^3/s$ for a 3-D implementation. This means that every term in the Lagrangian function Eq. (6.14) should have dimensions of m^5/s^3.

2: $\lambda_1^T g_1(u^{n+1})$:

$$\begin{aligned}\left[\lambda_1^T g_1(u^{n+1})\right]_{units} &\equiv \left[\phi\,(-Du^{n+1} + cbc)\right]_{units} \\ &= \left[\phi\,(Du^{n+1})\right]_{units} \\ &= (m^2/s^2)(m^2)(m/s) \\ &= m^5/s^3 \end{aligned} \tag{6.17}$$

This term is dimensionally consistent. We once again recall that the units for our Lagrange multiplier,

ϕ should be synonymous with pressure giving us units of m^2/s^2 when it is divided by density.

3: $\lambda_2^T g_2(u^{n+1})$:

$$\begin{aligned}\left[\lambda_2^T g_2(u^{n+1})\right]_{units} &\equiv \left[(f)^T W\ (Eu^{n+1} - u_B^{n+1})\right]_{units} \\ &= \left[(f)^T W\ (Eu^{n+1})\right]_{units} \\ &= (m^2/s^2)(m^2)(1)(m/s) \\ &= m^5/s^3 \end{aligned} \tag{6.18}$$

where W is a diagonal matrix that contains the surface areas of each boundary segment (for the u, v, and w lattices) and E is a dimensionless operator containing the multi-linear interpolation coefficients.

6.1.2 Derivation of a KKT System

The Lagrangian function that follows the new modified governing equations:

$$\begin{aligned}\mathcal{L}(u^{n+1}, \lambda_1, \lambda_2, \ldots, \lambda_M) &= f(u^{n+1}) + \sum_{k=1}^{M} \lambda_k^T g_k(u^{n+1}) \\ &= \frac{1}{2}\left(u^{n+1}\right)^T A\ u^{n+1} - \left(u^{n+1}\right)^T (r + mbc) \\ &+ \phi\ (-Du^{n+1} + cbc) \\ &+ f^T W\ (Eu^{n+1} - u_B^{n+1}) \end{aligned} \tag{6.19}$$

To find our KKT system, we differentiate the Lagrangian with respect to each of its variables and impose the first-order necessary condition for a feasible minimum to get:

$$\nabla_{(u^{n+1},\phi,f)}\mathcal{L}(u^{n+1}, \phi, f) = \begin{bmatrix} \nabla_{u^{n+1}}\mathcal{L}(u^{n+1}, \phi, f) \\ \nabla_{\phi}\mathcal{L}(u^{n+1}, \phi, f) \\ \nabla_{f}\mathcal{L}(u^{n+1}, \phi, f) \end{bmatrix} = 0$$

After performing the necessary derivations, we get:

$$\begin{bmatrix} Au^{n+1} - r - mbc - D^T\phi + E^T W^T f \\ \\ -Du^{n+1} + cbc \\ \\ WEu^{n+1} - Wu_B^{n+1} \end{bmatrix} = 0 \tag{6.20}$$

The above system can then be written in block matrix form as follows:

$$\begin{pmatrix} A & -D^T & E^T W^T \\ \\ -D & 0 & 0 \\ \\ WE & 0 & 0 \end{pmatrix} \begin{pmatrix} u^{n+1} \\ \\ \phi \\ \\ f \end{pmatrix} = \begin{pmatrix} r + mbc \\ \\ -cbc \\ \\ Wu_B^{n+1} \end{pmatrix} \tag{6.21}$$

Next we'll try to apply certain properties of our sub-matrices such as $D = -G^T$ and $WE = -(HW^T)^T$, to revert the above result for the KKT system back to our governing equations. We'll also multiply the second equation in Eq. (6.21) by -1. This leads to:

$$\begin{pmatrix} A & G & -HW^T \\ \\ D & 0 & 0 \\ \\ WE & 0 & 0 \end{pmatrix} \begin{pmatrix} u^{n+1} \\ \\ \phi \\ \\ f \end{pmatrix} = \begin{pmatrix} r + mbc \\ \\ cbc \\ \\ Wu_B^{n+1} \end{pmatrix} \tag{6.22}$$

which is exactly our original system (6.1)-(6.3). We have thus proven that our governing equations result in a KKT system.

6.1.3 Fractional Step Method

The fractional step method that we will employ to solve the system can be described by the following steps:

Step 1: Solve for the intermediate velocity u^*:

$$M\frac{u^* - u^n}{\Delta t} = \frac{\nu}{2}L(u^* + u^n) + mbc. \tag{6.23}$$

Step 2: Perform the projection by solving the following system:

$$\begin{pmatrix} DB^N G & -DB^N HW^T \\ EB^N G & -EB^N HW^T \end{pmatrix} \begin{pmatrix} \phi \\ f \end{pmatrix} = \begin{pmatrix} Du^* \\ Eu^* \end{pmatrix} + \begin{pmatrix} -cbc \\ u_B^{n+1} \end{pmatrix} \tag{6.24}$$

and then updating the velocity using:

$$u^{n+1} = u^* - B^N G\phi + B^N H W^T f. \tag{6.25}$$

6.2 Incremental-Lagrange Multiplier Projection Methods

Brown *et al.* [15] has presented quite a few pressure increment projection methods which are globally second-order accurate for the Navier-Stokes equations. We believe it is possible to extend these pressure increment schemes to the immersed boundary projection method by treating the pressure term as Lagrange multiplier. The downside to such an implementation is its difficulty to prove that such a scheme fits into a KKT system. In Perot's LU approximation, there's no such thing as a pressure update. Update equations for the Lagrange multipliers simply cannot be factored into the discrete operators that act on the velocity and Lagrange multipliers. As such it might simply be impossible to derive a KKT system for incremental-Lagrange multiplier projection methods. Fortunately, this will not affect the quality of such schemes or the possibility for a globally second-order projection method.

6.2.1 Fractional Step Method

The fractional step method used for the full IBPM system can be described by the following steps:

Step 1: Solve for the intermediate velocity u^*:

$$M\frac{u^* - u^n}{\Delta t} + Gp^{n-1/2} = \frac{\nu}{2}L(u^* + u^n) + HW^T f^{n-1/2} + mbc. \tag{6.26}$$

Step 2: Perform the projection by solving the following system:

$$\begin{pmatrix} \Delta t DM^{-1}G & -\Delta t DM^{-1}HW^T \\ \Delta t EM^{-1}G & -\Delta t EM^{-1}HW^T \end{pmatrix} \begin{pmatrix} \phi^{n+1} \\ \varphi^{n+1} \end{pmatrix} = \begin{pmatrix} Du^* \\ Eu^* \end{pmatrix} + \begin{pmatrix} -cbc \\ u_B^{n+1} \end{pmatrix}, \tag{6.27}$$

where the velocity is updated using:

$$u^{n+1} = u^* - \Delta t M^{-1}G\phi^{n+1} + \Delta t M^{-1}HW^T\varphi^{n+1}. \tag{6.28}$$

Step 3: Update the Lagrange multipliers:

$$Gp^{n+1/2} = Gp^{n-1/2} + G\phi^{n+1} - \frac{\nu\Delta t}{2} LM^{-1} G\phi^{n+1} \tag{6.29}$$

$$HW^T f^{n+1/2} = HW^T f^{n-1/2} + HW^T \varphi^{n+1} - \frac{\nu\Delta t}{2} LM^{-1} HW^T \varphi^{n+1} \tag{6.30}$$

Derivation of Lagrange Multiplier Updates

We rearrange the variables in Eq. (6.28) as follows:

$$u^* = u^{n+1} + \Delta t M^{-1} G\phi^{n+1} - \Delta t M^{-1} HW^T \varphi^{n+1} \tag{6.31}$$

and formulate the Lagrange multiplier updates by substituting u^* in Eq. (6.31) into Eq. (6.26) and comparing it to Eq. (6.1):

$$\begin{aligned} M\frac{u^{n+1}-u^n}{\Delta t} + \underbrace{Gp^{n-1/2} + G\phi^{n+1} - \frac{\nu\Delta t}{2} LM^{-1} G\phi^{n+1}}_{\approx Gp^{n+1/2}} = \\ \frac{\nu}{2} L(u^{n+1}+u^n) + \underbrace{HW^T f^{n-1/2} + HW^T \varphi^{n+1} - \frac{\nu\Delta t}{2} LM^{-1} HW^T \varphi^{n+1}}_{\approx HW^T f^{n+1/2}} \end{aligned} \tag{6.32}$$

Retrieving the primitive Lagrange multipliers from the equations above gives:

$$Gp^{n+1/2} = \underbrace{Gp^{n-1/2} + G\phi^{n+1} - \frac{\nu\Delta t}{2} LM^{-1} G\phi^{n+1}}_{RHS_1} \tag{6.33}$$

$$HW^T f^{n+1/2} = \underbrace{HW^T f^{n-1/2} + HW^T \varphi^{n+1} - \frac{\nu\Delta t}{2} LM^{-1} HW^T \varphi^{n+1}}_{RHS_2} \tag{6.34}$$

We can see from the previous equations, that the values for both $p^{n+1/2}$ and $f^{n+1/2}$ are not needed during the fractional step method, just the result of applying certain operators on them. Once again, if the Lagrange multipliers are needed at the end of the simulation (for visualization or testing purposes), we can take a divergence of Eqs. (6.33), (6.34) and invert the cell-centered discrete operators $DM^{-1}G$ and $DM^{-1}HW^T$.

$$DM^{-1}(Gp^{n+1/2}) = DM^{-1}(RHS_1) \tag{6.35}$$

$$DM^{-1}(HW^T f^{n+1/2}) = DM^{-1}(RHS_2) \tag{6.36}$$

6.3 Interpolation and Smoothing Functions

6.3.1 Multi-linear Interpolation

We first consider multi-linear interpolation for the E and H operators, with the E operator playing the role of interpolating face velocity to Lagrangian points and the H operator interpolates forces that lie on the Lagrangian points to cell faces. Differences with Taira and Colonius's E and H operators include:

No "smearing". The H operator in Taira and Colonius's implementation results in "smearing" of the singular boundary forces that lie on Lagrangian points over to a few neighboring cells. By replacing the discrete Dirac delta function with multi-linear interpolation, there is no smearing involved. Forces on the Lagrangian points merely interpolates to grid points within that same staggered cell. Hence the name "sharp" immersed boundary projection method.

Location of Lagrangian points. Taira and Colonius have Lagrangian points that are colocated. This means that the x, y, z components of the forces and velocities all lie on one same Lagrangian point. Our formulation will result in one set of Lagrangian points for each spatial direction. Lagrangian points are chosen to be the center of the immersed boundary segment in each of the staggered lattices.

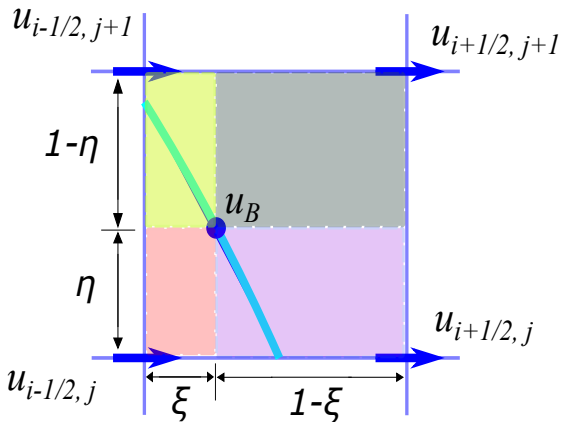
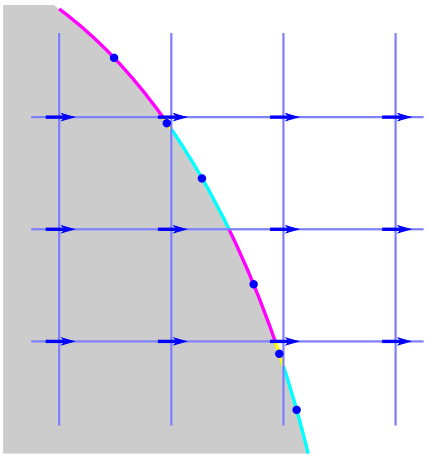


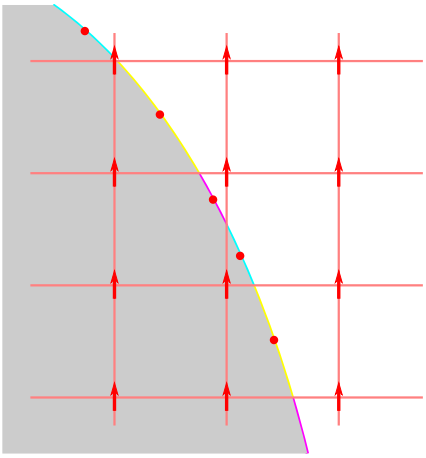
Figure 6.1: Bi-linear interpolation of u velocities to Lagrange point

Having proven in section 6.1.2 that for a KKT system to exist, $E = -H^T$, it would suffice to present the construction of E operator. As an illustrative example, we'll present the 2-D case in Figure 6.2.

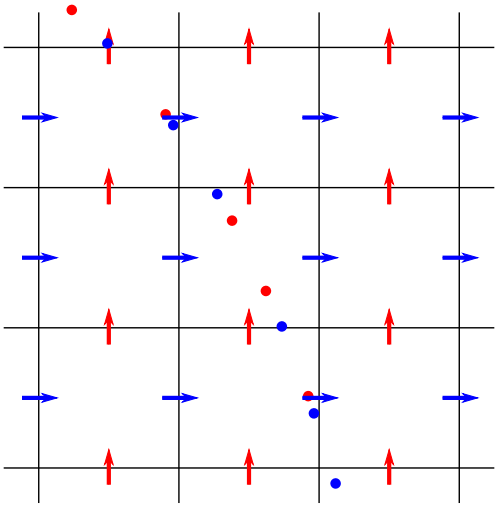
To shed some light on the details of the construction of the E operator, we will use a toy example looking into one of the x-staggered cell (see Figure 6.1).



(a) x staggered grid



(b) y staggered grid



(c) all Lagrangian unknowns

Figure 6.2: Lagrangian points for 2-D case

The above bilinear interpolation in the 2-D toy problem results in the following stencil for the E operator:

$$\begin{aligned} u_B = (1-\xi)\ \ (1-\eta) \quad & u_{i-1/2,j} \\ + \quad \xi \quad\ \ (1-\eta) \quad & u_{i+1/2,j} \\ + (1-\xi) \quad\ \ \eta \quad & u_{i-1/2,j+1} \\ + \quad \xi \quad\ \ \eta \quad & u_{i+1/2,j+1} \end{aligned} \tag{6.37}$$

While E is constructed using multi-linear interpolation, the H operator is constructed by means of the following relation: $H = -E^T$.

6.3.2 Higher-Order Interpolation and Smoothing

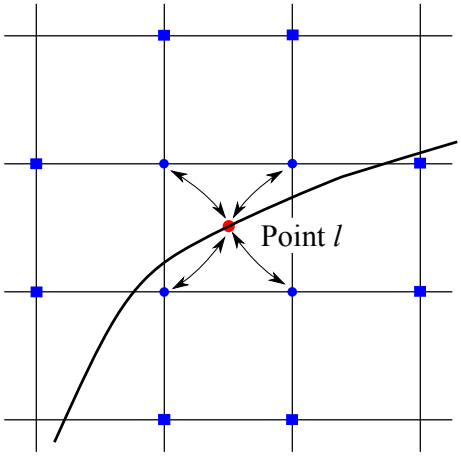


Figure 6.3: Effect of smoothing

In this section we attempt to generalize the formulation of the interpolation and smoothing functions and explain the possibility of achieving higher-orders.

Interpolation

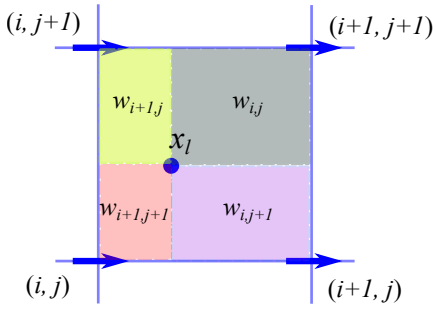


Figure 6.4: Interpolation weights

We refer to Figure 6.4 and rewrite the multi-linear interpolation for the two-dimensional case as

$$\varphi_f^l = \sum_{i,j} w_{ij}^l \varphi_{ij}, \tag{6.38}$$

where the weights necessarily have to be a convex sum for stability, i.e.

$$\sum_{i,j} w_{ij}^l = 1, \quad \forall l. \tag{6.39}$$

Smoothing

In the IB governing equations, information (in this case boundary forces) needs to be transferred from the Lagrangian points to the Eulerian grid, i.e.

$$\int_{\Delta s} \varphi_f(s)\ \mathrm{d}s = \int_{\Delta V} \varphi_{ij}(\mathbf{x})\ \mathrm{d}\mathbf{x}, \tag{6.40}$$

where Δs is the length between adjacent Lagrangian points, ΔV is the discrete volume. In the discrete 2-D case we rewrite the above formulation as

$$\varphi_{ij}^l = \sum_l \varphi_f^l w_{ij}^l \frac{\Delta s^l}{\Delta x_i \Delta y_j}. \tag{6.41}$$

We present the general two-dimensional function w_{ij}^l constructed by the product of single-variable functions,

$$\begin{gathered} w_{ij}^l(\mathbf{x}^l) = d\left(\xi\right) d\left(\eta\right), \\ \xi = \frac{x_f^l - x_i}{\delta x}, \qquad \eta = \frac{x_f^l - y_j}{\delta y}. \end{gathered} \tag{6.42}$$

In the case of multi-linear interpolation, the resulting smoothing function is

$$d : r \longmapsto \begin{cases} 1 + r & \text{if } -1 < r < 0 \\ 1 - r & \text{if } 0 < r < 1 \\ 0 & \text{otherwise} \end{cases}, \tag{6.43}$$

where r is the distance from Lagrangian point to Eulerian grid point.

Comments on Higher-Order Considerations

Multi-linear interpolation is exact for linear functions and second-order accurate for smooth functions. However, it is important to note that such interpolations when applied on true boundaries with zero thickness results in first-order accuracy. Peskin [43] noted that the use of multi-linear interpolation for true boundaries results in a velocity field which is not smooth. Although the velocity is continuous, it suffers a jump in the normal derivative. This is also true for the smoothing function in Eq. (6.43) which is a "hat" function.

Smoother kernels can be created to achieve higher-order accuracy. In particular, Peskin [43] gave some examples involving trigonometric functions. It should however be noted that kernels are required to satisfy a number of conditions. These conditions are elaborate and we refer the reader to Peskin [43] and Roma *et al.*

[27] for a complete list. In this sense, one simply need to change Eq. (6.43) to a smoother kernel (possibly with a wider stencil, see Figure 6.3) that satisfies all the required conditions to achieve matching accuracy with the temporal discretization of the IBPM.

Chapter 7

Conclusions and Future Work

The accuracy of projection methods for use in the Navier-Stokes equations has been studied theoretically and numerically in this thesis. The present investigation was motivated by the need to better understand the intricate interplay between the pressure term in the application of fractional step methods and its role as a Lagrange multiplier. The ability to obtain second or higher order temporal accuracy in pressure has given us the basis to extend this formulation to develop our sharp immersed boundary projection method.

In this thesis, we presented a new projection algorithm for the incompressible Navier-Stokes equations as well as the immersed boundary projection method. The algorithm is expected to resolve the problem of commutativity of discrete operators in the staggered grid by Harlow and Welch [20] such that the overall discretization is second-order accurate. The underlying motivation is to extend Brown *et al.*'s higher order pressure update schemes for the Navier-Stokes equations for use in immersed boundary projection methods. The pressure term in Navier-Stokes is most intuitively seen as a Lagrange multiplier and we subsequently demonstrated the global second-order property using normal mode analysis. The model is tested only on Navier-Stokes equations on a simple periodic box in 3-D to validate the second-order convergence of the method. The solutions appears to be quadratically convergent in both time and space even in the case of non-uniform grids.

Future work will provide numerical evidence for the sharp immersed boundary projection method proposed in this thesis. The present thesis has provided a clear quantitative description of the different classes of second-order projection methods. We proposed new ways in which Brown *et al.*'s approximate projection method can be extended to the staggered grid system. Indeed, a thorough understanding of higher order projection methods in the Navier-Stokes equations has proven critical to the development of the new sharp immersed boundary projection method which we believe improves upon current existing method [14].

In recent years, Sanderse and Koren in [40, 41] introduced a class of energy conserving time-integrators by means of implicit Runge-Kutta methods based on Gauss, Radau and Lobatto quadrature. These methods

are attractive in its application towards the solving of the incompressible Navier-Stokes equations due to its energy-conserving properties and most importantly, its promise towards higher order temporal accuracy for pressure. While classical fractional step methods seemed to have bottle-necked at global second-order accuracy, Sanderse and Koren saw the incompressible Navier-Stokes equations as a set of differential-algebraic equations (DAE) to which Runge-Kutta methods can be applied. Sanderse *et al.* did some accuracy studies on both explicit [42] and implicit [40] Runge-Kutta methods and we will briefly summarize these recent findings. For explicit Runge-Kutta methods, it was noted that if boundary conditions for the continuity equations are independent of time, one can achieve same order of accuracy in both pressure and velocity with respect to the "classical" order of the Runge-Kutta method at the cost of an additional Poisson solve. However, if continuity boundary conditions are time-dependent, Sanderse showed that it is possible to construct higher-order approximations by linear combinations of stage values of pressure. In this approach, one cannot expect to attain the same order of accuracy for pressure as the velocity. Sanderse proposed some ways to achieve second-order pressure in [42] and seemed to have come to the conclusion that although higher order accurate pressures might be attainable by means of more stages, such methods are not very relevant from a practical point of view. They concluded that the "best" explicit Runge-Kutta method for the incompressible Navier-Stokes is a three-stage method of type M2, which results in third-order velocity and second-order for pressure.

Energy conserving methods are attractive because they are free of numerical diffusion which could possibly overwhelm molecular diffusion in turbulent flow simulations. This ensures that all modeled diffusion are not artificial. Such schemes are also not limited by restrictive time steps and are useful for simulations which take place in large time or length scales. These energy-conserving methods [41] which are necessarily implicit, requires solving an additional Poisson problem in order for pressure to be of same order as velocity regardless of steady or time-dependent continuity boundary conditions. One of these schemes is the diagonally implicit Runge-Kutta (DIRK) method which has the advantage that the stages of the Runge-Kutta method can be solved sequentially (due to zeros in upper triangle of the Butcher tableau). Nevertheless, with all the advantages of the energy-conserving schemes, it is required to solve a non-linear fully coupled saddle point problem in every stage per time-step. This is extremely costly and the trade-off between stability and cost have to be carefully considered.

An important difference between (1) the standard second-order pressure projection methods (i.e. Kim and Moin [13] and Bell, Colella and Glaz [8]) and (2) Sanderse's Runge-Kutta schemes [40, 41, 42] is that in order to achieve global second-order accuracy, Sanderse requires the solution of 2 Poisson problems, while

the former only require one. Since most of the computational time is actually spent on the solution of the Poisson system, Sanderse's method could substantially be more costly. Nevertheless, Sanderse's work is more general, and in particular he extends it to 4th order relatively seamlessly. This raises the following possibility: could Sanderse's method be combined with classical second-order projection methods to get for example, a global 3rd order scheme that requires the solution of only 2 Poisson systems (or perhaps global 4th order scheme with 3 Poisson systems)? In the near future, we would like to incorporate Brown *et al.*'s [15] incremental-pressure projection methods into Sanderse's work, by including pressure terms (i.e. gradient of pressure) into the intermediate stages of Runge-Kutta schemes in the hope of achieving higher-order schemes with fewer Poisson problems to be solved.

These Runge-Kutta methods seem to be the future of achieving higher-orders in the solving of the incompressible Navier-Stokes equations. For implicit methods, computational cost is restricted by ability to solve the non-linear saddle point problem efficiently, while the most practical explicit scheme recommended by Sanderse only leads to second-order pressure. At this point, such methods do not seem competitive to classical fractional step methods [15, 4, 13, 8, 9]. As research in this field progresses and cheaper methods developed, we do foresee the use of such higher-order schemes in our future implementation.

Bibliography

[1] C.M. Rhie, and W.L. Chow, *A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation*, AIAA J., Vol. 21. 11, pp. 1525-1532 (1983).

[2] Y. Morinishi, T.S. Lund, O.V. Vasilyev, and P. Moin, *Fully conservative higher order finite difference schemes for incompressible flow*, J. Comput. Phys., Vol. 143, pp. 90-124 (1998).

[3] F.N. Felten and T.S. Lund, *Critical comparison of the colocated and staggered grid arrangements for incompressible turbulent flows*, Proceedings of the Third AFOSR International Conference on DNS/LES (2001).

[4] J.B. Perot, *An Analysis of the fractional step method*, J. Comput. Phys. 108, pp. 51-58 (1993).

[5] J.B. Perot, *Letter to the editor*, J. Comput. Phys. 121, pp. 190-191 (1995).

[6] S. Abdallah, *Comments on the fractional step method*, J. Comput. Phys. 117, pp. 179 (1995).

[7] K. Kim, S. Baek, H. Sung, *An implicit velocity decoupling procedure for the incompressible Navier–Stokes equations*, International Journal for Numerical Methods in Fluids. 38, pp. 125–138 (2001).

[8] J.B. Bell, P. Colella, and H.M. Glaz, *A second order projection method for the incompressible Navier–Stokes equations*, J. Comput. Phys. 85, pp. 257 (1989).

[9] J.B. Bell, P. Colella, and L.H. Howell, *An efficient second-order projection method for viscous incompressible flow*, in Proceedings of the Tenth AIAA Computational Fluid Dynamics Conference, AIAA, pp. 360 (1991).

[10] J.C. Strikwerda and Y. S. Lee, *The accuracy of the fractional step method*, SIAM J. Numer. Anal. 37, pp. 37 (1999).

[11] W. E and J. Guo Liu, *Projection method. I. Convergence and numerical boundary layers*, SIAM J. Numer. Anal. 32, pp. 1017 (1995).

[12] J.C. Strikwerda, *Finite difference schemes and partial differential equations, 2nd edition*, SIAM, Chapter 11 (2004).

[13] J. Kim and P. Moin, *Application of a fractional-step method to incompressible Navier–Stokes equations*, J. Comput. Phys. 59, pp. 308 (1985).

[14] K. Taira and T. Colonius, *The immersed boundary method: A projection approach*, J. Comput. Phys. 225, pp. 2118-2137 (2007).

[15] D. Brown, R. Cortez, and M. Minion, *Accurate projection methods for the incompressible Navier–Stokes equations*, J. Comput. Phys. 168, pp. 464–499 (2001).

[16] M.M. Rai and P. Moin, *Direct simulations of turbulent flow using finite-difference schemes*, J. Comput. Phys. 96, pp. 15-53 (1991).

[17] R.M. Beam and R.F. Warming, *An Implicit Factored Scheme for the Compressible Navier–Stokes Equations*, AIAA Journal, Vol. 16, No. 4, pp. 393-402 (1978).

[18] G. Tryggvason , *Lecture 25: Elementary Grid Generation* , Lecture Notes from University of Notre Dame for Computational Fluid Dynamics (2011).

[19] M. Rosenfeld, *An uncoupled temporally second-order accurate implicit solver of the incompressible Navier-Stokes equations*, AIAA Journal, Vol. 34, pp. 1829-1834 (1996).

[20] E H. Harlow and J.E. Welch, *Numerical calculation of time-dependent viscous incompressible flow of fluids with free surfaces*, Phys. Fluids, 8, pp. 2182-2189 (1965).

[21] A.S. Almgren, J.B. Bell and W.G. Szymczak, *A numerical method for the incompressible Navier-Stokes equations based on an approximate projection*, SIAM J. Sci. Comput. 17(2), (1996).

[22] M.F. Lai, J. Bell and P. Colella, *A projection method for combustion in the zero Mach number limit*, in Proceedings of the Eleventh AIAA CFD Conference, AIAA, pp. 776 (1993).

[23] M.L. Minion, *A projection method for locally refined grids*, J. Comput. Phys. 127, pp. 158–178 (1996).

[24] A.J. Chorin, *Numerical solution of the Navier-Stokes equations*, J. Math. Comput., 22, pp. 745-762 (1968).

[25] J. van Kan, *A second-order accurate pressure-correction scheme for viscous incompressible flow*, SIAM J. Sci. Comput. 7, 870 (1986).

[26] L.H. Howell, *A multilevel adaptive projection method for unsteady incompressible flow*, 6th Copper Mountain Conf. on Multigrid Methods (1993).

[27] A.M. Roma, C.S. Peskin and M.J. Berger, *An adaptive version of the immersed boundary method*, J. Comput. Phys. 153, 509-534 (1999).

[28] A.J. Chorin, *On the convergence of discrete approximations to the Navier-Stokes equations*, J. Math. Comput., 23, pp. 341-353 (1969).

[29] K. Taira, *The immersed boundary projection method and its application to simulation and control of flows around low-aspect-ratio wings*, CalTech Thesis (2008).

[30] T. Colonius and K. Taira, *A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions*, Comput. Methods Appl. Mech. Engrg. 197, 2131-2146 (2008).

[31] W. Chang, F. Geraldo and B. Perot, *Analysis of an exact fractional step method*, J. Comput. Phys. 180, 183-199 (2002).

[32] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge Univ. Press (2009).

[33] W.D. Henshaw, *A fourth-order accurate method for the incompressible Navier–Stokes equations on overlapping grids*, J. Comput. Phys. 113, pp. 13-25 (1994).

[34] CS. Peskin, *Flow patterns around heart valves: a digital computer method for solving the equations of motion*, PhD thesis. Physiol., Albert Einstein Coll. Med., Univ. Microfilms (1972).

[35] D. Goldstein, R. Handler and L. Sirovich, *Modeling a no-slip flow boundary with an external force field*, J. Comput. Phys. 105, pp. 354–366 (1993).

[36] R.P. Beyer and R.J. LeVeque, *Analysis of a one-dimensional model for the immersed boundary method*, SIAM J. Numer. Anal. 29, pp. 332-364 (1992).

[37] R. Mittal and G. Iaccarino, *Immersed boundary methods*, Annu. Rev. Fluid Mech. 37, pp. 239–61 (2005).

[38] U. Ascher and L. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*, SIAM (1998).

[39] E. Hairer, C. Lubich and M. Roche, *The numerical solution of differential-algebraic systems by Runge-Kutta methods*, Springer (1989).

[40] B. Sanderse and B. Koren, *Runge-Kutta methods for the incompressible Navier-Stokes equations*, 21st AIAA Computational Fluid Dynamics Conference, 3085 (2013).

[41] B. Sanderse, *Energy-conserving Runge-Kutta methods for the incompressible Navier-Stokes equations*, J. Comput. Phys. 233, pp. 100-131 (2013).

[42] B. Sanderse and B. Koren, *Accuracy analysis of explicit Runge–Kutta methods applied to the incompressible Navier–Stokes equations*, J. Comput. Phys. 231, pp. 3041–3063 (2012).

[43] C.S. Peskin, *The immersed boundary method*, Acta Numerica, 11, pp. 479–517 (2003).